Um sistema de processamento escalável de consultas analíticas sobre data *warehouses* criptografados e armazenados na nuvem

Claudivan Cruz Lopes [1], Valéria Cesário Times [2]

[1] claudivan@ifpb.edu.br. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – Campus Patos. [2] vct@cin.ufpe.br . Universidade Federal de Pernambuco – Centro de Informática.

RESUMO

Sistemas de processamento de consultas para bancos de dados criptografados têm sido propostos como uma solução para proteger a confidencialidade de dados armazenados em servidores não confiáveis, tais como provedores de computação em nuvem, e para reduzir o impacto da criptografia no desempenho de consultas sobre dados criptografados. Pouca atenção, no entanto, tem sido dedicada em determinar como a escalabilidade provida pela computação em nuvem pode ser usada para melhorar o desempenho de consultas analíticas sobre Data Warehouses (DW) criptografados. Nesse sentido, este artigo descreve um sistema de processamento escalável de consultas analíticas sobre DW criptografados armazenados na nuvem, em que são especificados os componentes arquiteturais que dão suporte a esta escalabilidade.

Palavras-chave: Data Warehouses Criptografados. Computação em Nuvem. Arquiteturas de Sistemas.

ABSTRACT

Query processing systems for encrypted databases has been proposed as a solution to protect the confidentiality of data stored in untrusted servers, such as cloud computing providers, and to decrease the impact of encryption on query performance over encrypted data. However, little attention has been devoted in determining how the scalability provided by cloud computing can be used to increase the performance of analytical queries over encrypted Data Warehouses (DW). In this sense, this article describes a system for scalable processing of analytical queries over encrypted DW stored in the cloud, where the system architectural components that provide support to such scalability are specified.

Keywords: *Encrypted Data Warehouses. Cloud Computing. Systems Architectures.*

1 Introdução

A crescente quantidade de dados sigilosos armazenados em bancos de dados tem tornado a confidencialidade de dados fundamental para as organizações. De fato, estudos recentes relatam um grande número de incidentes provocados por vazamento de diversos tipos de dados, tais como pessoais, financeiros, detalhes sobre seguros e cartões de crédito (KOUNS; MARTIN, 2015).

O risco de violação da confidencialidade de dados é especialmente latente quando estes são armazenados em provedores de computação em nuvem, tais como provedores de *Database as a Service* (DaaS), já que os dados ficam residentes nas instalações do próprio provedor, as quais são normalmente ambientes desconhecidos da maioria dos usuários (SRINIVASAMURTHY *et al.*, 2013).

Uma alternativa para manter a confidencialidade de dados na nuvem é o uso da criptografia, recurso que permite que dados sigilosos sejam criptografados num ambiente seguro do usuário antes de enviá-los à nuvem para serem armazenados (GOSAIN, ARORA, 2015; LOPES, TIMES, 2015). Consultar dados criptografados requer, porém, sua decodificação, fato que pode comprometer a confidencialidade de dados se a decodificação é feita no próprio provedor de DaaS, ou pode acarretar um alto custo de processamento, se todos os dados criptografados forem transferidos para o ambiente do usuário, para serem decodificados e as consultas serem executadas e concluídas.

Como forma de proteger a confidencialidade de dados e reduzir o impacto da criptografia no tempo de execução de consultas, estudos propõem sistemas para processar consultas sobre dados criptografados, os quais usam técnicas de criptografia que permitem que operações sejam realizadas sobre os dados ainda criptografados (HACIGUMUS, IYER, MEHROTRA, 2004; TU *et al.*, 2013; LOPES *et al.*, 2014; LIU, 2014; POPA, 2014; BABY, CHERUKURI, 2015; SMITH *et al.*, 2014). Desse modo, as consultas são executadas sem precisar decodificar dados na nuvem.

Sob outra perspectiva, a confidencialidade de dados e o desempenho de consultas são questões importantes em *Data Warehouses* (DW) mantidos na nuvem, pois um DW armazena dados de uso gerenciais e estratégicos para uma empresa, além de que as consultas analíticas feitas sobre um DW requerem alto grau de desempenho (SANTOS, BERNARDINO, VIEIRA, 2011; SANTOS *et al.*, 2013). Por conseguinte, a implantação de um *DW criptografado* (DWC) na nuvem pode contribuir para a proteção da confidencialidade de dados e possibilitar ganhos de desempenho referentes ao processamento de consultas analíticas sobre dados criptografados, uma vez que o grande volume de dados de um DW e a própria execução de consultas analíticas podem ser escalados na infraestrutura fornecida por provedores de computação em nuvem. De acordo com o levantamento do estado da arte, porém, os sistemas de processamento de consultas sobre dados criptografados propostos têm limitações para usufruir dessa escalabilidade, pois não foram projetados para dar suporte a esta escalabilidade.

A especificação de um sistema de processamento escalável de consultas analíticas sobre DWC armazenados na nuvem constitui, portanto, o foco principal deste artigo, cujas contribuições são colocadas a seguir:

- É descrita uma revisão bibliográfica sobre técnicas de criptografia e suas funcionalidades no processamento de dados criptografados;
- É apresentado o resultado de uma revisão do estado da arte sobre sistemas de processamento de consultas sobre bancos de dados criptografados;
- É especificado um sistema de processamento escalável de consultas analíticas sobre DWC armazenados na nuvem e seus componentes arquiteturais necessários para dar suporte a essa escalabilidade.

O restante deste artigo está organizado da seguinte forma: na Seção 2 são resumidos os principais conceitos usados neste artigo, enquanto a Seção 3 aborda sistemas de processamento de consultas sobre bancos de dados criptografados. A Seção 4 apresenta o sistema proposto e a Seção 5 conclui este artigo e enumera trabalhos futuros.

2 Fundamentação teórica

Nesta seção é descrita a fundamentação teórica que aborda conceitos sobre DW, consultas analíticas e conceitos e características da computação em nuvem. Por fim, são relatadas técnicas de criptografia e suas aplicações no processamento de consultas.

2.1 DW e consultas analíticas

Um DW é uma grande base de dados não volátil, integrada e corporativa, cujos dados são orientados por assuntos de negócio, variantes no tempo e históricos, e usados exclusivamente para consultas e recuperação de informações de suporte à decisão de nível gerencial (INMON, 2005).

Por *integrados* entende-se que os dados de um DW possuem formato homogêneo e consistente, integrados numa única base de dados; *orientados por assunto* significa que um DW contém dados agrupados por assuntos de interesse de um negócio em questão, p.ex. vendas, compras e produção; *não volátil* implica que, uma vez que os dados são incluídos num DW, estes não são mais excluídos, porém, excepcionalmente, podem ser modificados no caso de correção da carga dos dados; e *variantes no tempo* e *históricos* implica que os dados de um DW representam a consolidação dos fatos de negócio no momento em que eles aconteceram.

Tipicamente, um DW é representado por um *modelo dimensional* (KIMBALL; ROSS, 2013) que possibilita eficiência na organização dos dados e na recuperação de informações gerenciais. O modelo dimensional é baseado nos conceitos de *fato*, *dimensão* e *medida*. Um fato representa um assunto de negócio a ser analisado, enquanto uma dimensão representa uma perspectiva de visualização de um fato. Uma medida, por sua vez, é um valor numérico que quantifica um fato.

Quanto ao processamento de consultas, ferramentas OLAP (*OnLine Analytical Processing*) são as principais interfaces para consultas aos dados de um DW (CODD; COOD; SALLEY, 1993). Estas ferramentas reconhecem a natureza dimensional dos dados de um DW e providenciam funcionalidades para sua navegação, permitindo que gerentes executem consultas analíticas com alto desempenho e interatividade e que projetem os dados consultados a partir de múltiplas visões.

Conforme ilustrado na Figura 1, consultas analíticas tipicamente requerem o processamento de funções de agregação sobre medidas (p.ex. a soma) e o uso de restrições de seleção sobre os dados dimensionais de um DW, em que os resultados são agrupados e ordenados por operadores de ordenação e agrupamento.

Em relação à carga de dados, ferramentas de *Extração, Transformação e Carga (ETL)* são utilizadas para extrair dados de fontes de dados externas ao DW e transformá-los em dados padronizados para inseri-los num DW (KIMBALL; ROSS, 2013).

Figura 1 – Exemplo de uma consulta analítica típica

```
SELECT Col1, Col2, ...., ColN, SUM(Medida)
FROM Tabela de Fatos INNER JOIN Tabelas de dimensão
WHERE Restrições
GROUP BY Col1, Col2, ...., ColN
ORDER BY Col1, Col2, ...., ColN
```

Fonte: O autor.

2.2 Computação em nuvem

Computação em nuvem é um modelo de computação em que *hardware* (armazenamento, memória e CPU), infraestrutura de rede e *software* (servidores, sistemas operacionais e aplicações) são serviços disponibilizados por meio da *Internet,* por centros de dados especializados chamados *provedores*, e acessados pelos usuários de acordo com a necessidade de suas demandas (MELL; GRANCE, 2011).

Os serviços em nuvem são *configuráveis*, ou seja, o usuário escolhe os recursos de rede, *hardware* e *software* que deseja utilizar; esses recursos são *medidos* pelo provedor e são pagos pelo usuário proporcionalmente ao seu consumo; também são *elásticos* e *escaláveis*, pois podem ser adquiridos, liberados, diminuídos ou aumentados de forma dinâmica, exigindo pouco esforço gerencial ou pouca interação com o provedor (BADGER *et al.*, 2012).

Um dos serviços providos na computação em nuvem é conhecido como *Database as a Service* (DaaS) (HACIGUMUS *et al.* 2002). O DaaS transfere as tarefas complexas de administração e *tuning* dos sistemas de gerenciamento de bancos de dados (SGBD) para o provedor, permitindo que os usuários abstraiam essa complexidade e se concentrem na modelagem, implantação e acesso às suas bases de dados.

Quanto às tecnologias de gerenciamento de dados na nuvem, alguns estudos investigaram a implantação de SGBD na nuvem. Sousa, Moreira e Machado (2009) classificaram SGBD em nuvem como: não relacionais e nativos para a nuvem, p.ex. *Hbase*; não relacionais e não nativos para a nuvem, mas que são executados na nuvem, p.ex. *MongoDB*; relacionais e nativos para a nuvem, p.ex. *Windows Azure SQL Database*; e relacionais e não nativos para a nuvem, mas que são utilizados na nuvem, p.ex. *Amazon Relational Database Service*.

Por outro lado, Abadi (2009) investigou a implantação de SGBD transacionais e analíticos na nuvem e concluiu que SGBD analíticos, tais como sistemas de DW, são mais adequados para implantação na nuvem em comparação aos SGBD transacionais, em função de suas arquiteturas permitirem a execução de consultas com alto grau de escalabilidade e por não requererem garantias da propriedade de consistência de transações. Abadi (2009) ressaltou, contudo, a necessidade da criação de mecanismos para garantir a confidencialidade de dados armazenados na nuvem, já que provedores de DaaS são ambientes não confiáveis sob a perspectiva do usuário (THOMPSON *et al.*, 2009). Diante disso, mecanismos de proteção à confidencialidade de dados mantidos na nuvem têm sido propostos, entre os quais se pode destacar o uso de técnicas de criptografia.

2.3 Técnicas de criptografia

Técnicas de criptografia devem permitir que operações sejam feitas sobre dados criptografados, com o intuito de proteger a confidencialidade de dados e reduzir o impacto da criptografia no processamento de dados criptografados.

É importante considerar que essas técnicas têm um nível particular de funcionalidade e de segurança: a funcionalidade indica as operações que podem ser realizadas sobre os dados criptografados; e a segurança indica o tipo de informação sobre os dados criptografados que é revelado ao provedor. Por exemplo, a *criptografia simétrica determinística* (CSD) permite a comparação de dados criptografados por meio do uso de operadores de igualdade, porém revela a duplicidade de dados criptografados (KADHEM; AMAGASA; KITAGAMA, 2009). A *Criptografia de preservação de ordem* (CPO) possibilita que dados criptografados possam ser ordenados e comparados por meio de operadores relacionais, mas revela a ordem e a duplicidade dos dados criptografados (POPA, LI, ZELDOVICH, 2013; CHUNG, OZSOYOGLU, 2006). A *Criptografia multivalorada de preservação da ordem* (CMPO) é uma técnica CPO em que os dados criptografados são distintos entre si, mas que ainda revelam sua ordem (KADHEM; AMAGASA; KITAGAMA, 2013). A *Criptografia aditivamente homomórfica* (CAH) permite o cálculo de soma sobre dados criptografados e tem segurança probabilística, provendo indistinguibilidade de dados criptografados (POPA *et al.*, 2012). Já a *Criptografia de pesquisa difusa* (CPD) permite a realização de busca textual sobre dados alfanuméricos criptografados e tem segurança probabilística (WU *et al.* 2012). Por fim, há ainda a técnica de *Criptografia baseada em partição* (CBP), que ordena e divide um domínio em partições identificadas por valores aleatórios distintos. Cada valor do domínio é criptografado usando uma técnica CSD e mantido junto com o identificador da partição correspondente. Este identificador serve como um índice que permite a comparação de identificadores por meio do uso de operadores relacionais e de igualdade (HORE *et al.*, 2012).

O uso de técnicas de criptografia sobre bancos de dados viabiliza o processamento de alguns tipos de consultas. A Figura 2 ilustra técnicas de criptografia e os tipos de consultas possíveis, em que *att* corresponde a um atributo do banco de dados, enquanto *E* e *MAP* equivalem às funções de encriptação. Com base no uso dessas técnicas, sistemas de processamento de consultas sobre bancos de dados criptografados têm sido propostos. A seguir é mostrada uma investigação sobre alguns desses sistemas.

Figura 2 – Técnicas de criptografia e tipos de consultas possíveis

Técnica de Criptografia	Tipo de Consulta	Exemplo
CSD	Consultas com restrições de igualdade	SELECT ... WHERE ***att*** = ***E***(10)...
	Consultas com agrupamentos de dados	SELECT ... GROUP BY ***att***...
CPO	Consultas com ordenação de dados	SELECT ... ORDER BY ***att***...
	Consultas com agrupamentos de dados	SELECT ... GROUP BY ***att***...
	Consultas com função de máx. e/ou mín.	SELECT ... MAX(***att***)...
	Consultas com restrições por intervalo	SELECT ... WHERE ***att*** ≥ ***E***(10)...
CMPO	Consultas com ordenação de dados	SELECT ... ORDER BY ***att***...
	Consultas com função de máx. e/ou mín.	SELECT ... MIN(***att***)...
	Consultas com restrições por intervalo	SELECT ... WHERE ***att*** < ***E***(10)...
CAH	Consultas com função de soma	SELECT ... SUM(***att***)...
CPD	Consultas com busca textual	SELECT ... WHERE ***att*** LIKE ***E***(10)...
CBP	Consultas com restrições de igualdade	SELECT ... WHERE ***att*** = ***MAP***(10)...
	Consultas com restrições por intervalo	SELECT ... WHERE ***att*** IN ***MAP***(10)...

Fonte: O autor.

3 Sistemas de processamento de consultas sobre bancos de dados criptografados

Sistemas de processamento de consultas sobre bancos de dados criptografados fazem uso de uma combinação de técnicas de criptografia para permitir a execução de consultas mais complexas sobre dados criptografados do que as ilustradas na Figura 2. Hacigumus, Iyer e Mehrotra (2004) propuseram um sistema OLTP que usa técnicas CBP, CSD e CAH para permitir a execução de consultas com restrições de seleção, agrupamento e função de soma sobre dados criptografados. Nesse sistema, cada item de dado é classificado como *aggregation*, *field-level* ou *parti-*

tioning, sendo assim criptografado por uma técnica CAH, CSD ou CBP, respectivamente. Também é criado um atributo *etuple* em cada registro de dados, o qual equivale ao valor criptografado de todos os itens de dados do próprio registro e é usado na projeção de resultados de consultas.

Liu (2014) propôs um sistema OLTP em que cada item de dado é criptografado por uma tripla de técnicas: CAH, um mecanismo CMPO e uma função *hash* segura (KIM *et al.*, 2006). Assim, a encriptação de um item de dado produz três valores distintos, mantidos em colunas sufixadas por *Enc*, *Rng* e *Eq*, respectivamente. Com isso, o sistema permite a realização de consultas com restrições de seleção, ordenação, agrupamento e funções de agregação sobre dados criptografados, contudo não suporta ordenação e agrupamento numa mesma consulta, já que estas operações devem ser executadas sobre as colunas *Rng* e *Eq*, respectivamente, enquanto os SGBD exigem que sejam executadas sobre as mesmas colunas.

Popa *et al.* (2012) propuseram o *CryptDB*, um sistema OLTP que provê uma criptografia ajustável chamada *cebola*, para dar suporte a múltiplas computações sobre dados criptografados. *CryptDB* produz uma ou mais cebolas para cada item de dado, de modo que cada cebola é formada por várias camadas de criptografia. Cada camada de uma cebola aplica uma técnica de criptografia específica sobre um item de dado, a qual é baseada em CSD, CPO, CAH ou CDP. O conjunto de cebolas provido por *CryptDB* permite a execução de consultas com busca textual, restrições de seleção, ordenação, agrupamento e funções de agregação sobre os dados criptografados.

Tu *et al.* (2013) desenvolveram um sistema de processamento de consultas analíticas sobre bancos de dados criptografados chamado *Monomi*. Este sistema contém um *designer* que determina as técnicas de criptografia a serem aplicadas aos dados em função do esquema do banco de dados do usuário e de um conjunto-exemplo de dados e consultas. O esquema de dados resultante é composto por atributos cujos dados são encriptados por técnicas CSD, CPO, CAH e CDP. *Monomi* também aplica um otimizador de consultas que introduz técnicas para acelerar o processamento de consultas analíticas. Essas técnicas são: a *per-row precomputation*, que materializa os resultados encriptados de certas computações sobre os itens de dado; e a *conservative pre-filtering*, a qual materializa os resultados de operações que não podem ser executadas sobre alguns itens de dados criptografados.

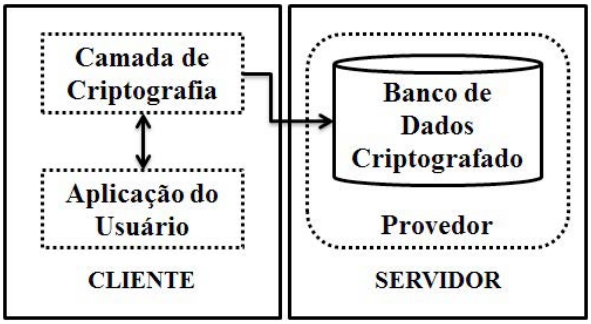
Lopes *et al.* (2014) propuseram um sistema de processamento de consultas analíticas sobre DWC armazenados na nuvem. Este sistema se baseou em análises experimentais de desempenho de consultas analíticas para definir a aplicação de múltiplas técnicas de criptografia em que atributos descritivos de tabelas de dimensão e de fatos de um DW são criptografados por uma técnica CMPO, e medidas são criptografadas por técnicas CMPO e CAH. Com isso, restrições de seleção, funções de soma, ordenação e agrupamentos definidos em consultas analíticas são feitos sobre os dados dimensionais criptografados.

Em geral, tais sistemas definem uma arquitetura cliente-servidor como ilustrada na Figura 3. O cliente corresponde ao ambiente do usuário e é considerada a parte menos vulnerável da arquitetura; é composto pela aplicação do usuário, que define as interfaces de acesso aos dados, e pela camada de criptografia, que permite o uso transparente da criptografia para o usuário. Esta camada tem as responsabilidades de criptografar os dados antes de armazená-los no servidor, traduzir as consultas de modo a serem executadas sobre os dados criptografados mantidos no servidor, decodificar os dados recuperados do servidor e gerenciar as chaves de criptografia. Já o servidor engloba um banco de dados criptografado mantido num provedor não confiável, tal como um provedor de DaaS, que somente tem acesso aos dados criptografados.

Apesar das contribuições relevantes na área de consultas sobre dados criptografados, tais sistemas têm deficiências na provisão da escalabilidade no processamento de consultas, pois suas arquiteturas não proveem suporte à escalabilidade (POPA *et al.*, 2012; HACIGUMUS, IYER, MEHROTRA, 2004; LIU, 2014; TU *et al.*, 2013) ou sofrem limitações na provisão da escalabilidade, com uma arquitetura na qual um único orquestrador é responsável pela execução de consultas (LOPES *et al.*, 2014). Um orquestrador é um componente que coordena todo o processamento no sistema. Assim, um sistema com um único orquestrador pode ficar inoperante se este falhar, além de que não é capaz de paralelizar o processamento de consultas simultâneas.

Na próxima seção é especificado um sistema que supre as limitações supracitadas ao definir uma arquitetura escalável para a manipulação dos dados de um DWC mantidos na nuvem.

Figura 3 – Arquitetura cliente-servidor para o processamento de dados criptografados



Fonte: O autor.

4 O sistema proposto

Nesta seção são apresentados os aspectos de análise e projeto do sistema proposto. A Seção 4.1 descreve seus requisitos e funcionalidades. As Seções 4.2 e 4.3 descrevem, respectivamente, a arquitetura e o esquema de metadados do sistema proposto. A Seção 4.4 apresenta o módulo de criptografia, o qual permite a extensibilidade do sistema em relação às técnicas de criptografia aplicados sobre um DW. A Seção 4.5 discorre sobre o modelo de ameaças ao sistema, enquanto a Seção 4.6 descreve as limitações do sistema.

4.1 Requisitos e funcionalidades

Para executar consultas analíticas sobre DWC armazenados na nuvem, um usuário administrador define como a criptografia deve ser aplicada sobre um DW em particular. Isto inclui a definição das regras empregadas para mapear o esquema lógico do DW num esquema lógico do DWC correspondente; a especificação de quais e como as operações requeridas em consultas analíticas devem ser executadas sobre um DWC; a definição das regras utilizadas para reformular as consultas analíticas do usuário de modo a serem executadas sobre um DWC; e a definição das técnicas de criptografia empregadas bem como suas funções de encriptação e de decodificação de dados.

Para salvaguardar a confidencialidade de dados, o ambiente do usuário cumpre as seguintes atribuições no sistema: grava os esquemas lógicos do DW e do DWC correspondente; criptografa os nomes de tabelas e atributos de um DWC antes de criar o esquema físico num provedor, mantendo a semântica associada aos nomes confidencial; reformula consultas analíticas do usuário para habilitar sua execução sobre um DWC e para evitar que um provedor conheça o mapeamento de esquemas que foram aplicados; realiza os processos de criptografia e decodificação de itens de dados; e mantém as chaves de criptografia utilizadas nesses processos.

Para permitir a escalabilidade no armazenamento de dados e na execução de consultas analíticas, o sistema particiona os dados de um DWC entre vários *hosts* na nuvem, no qual o processamento de consultas é feito por processadores distribuídos que executam em paralelo e independentes. Isso possibilita a execução simultânea de consultas analíticas e previne que o sistema fique inoperante quando algum processador falhar.

Sob o ponto de vista do usuário, o sistema proposto apresenta as funcionalidades enumeradas a seguir.

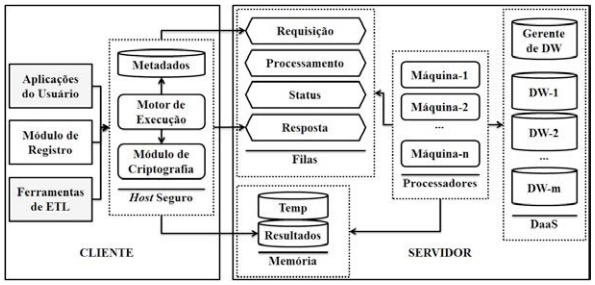
1. *Cadastrar mapeamentos e esquemas*. O sistema armazena as regras que são usadas para mapear o esquema lógico de um DW num esquema lógico do DWC correspondente, as regras utilizadas para mapear as operações de consultas analíticas e os próprios esquemas lógicos do DW e do DWC;
2. *Registrar chaves de criptografia*. O sistema obtém e armazena as chaves usadas na criptografia de nomes de tabelas e atributos de um DWC, assim como as chaves utilizadas na criptografia e na decodificação de itens de dado de um DW e de parâmetros e resultados de consultas;
3. *Incluir dados*. O sistema criptografa os dados e os armazena em algum *host* na nuvem;
4. *Executar consultas analíticas*. O sistema processa as consultas analíticas do usuário. Isso envolve a reformulação de consultas e a criptografia dos parâmetros definidos nas consultas, a execução distribuída de consultas na nuvem e a decodificação dos resultados obtidos.

4.2 Arquitetura do sistema

A Figura 4 ilustra a arquitetura do sistema proposto, a qual estende a arquitetura apresentada na Figura 3. O cliente equivale ao ambiente do usuário e compreende os componentes: *módulo de registro*, *ferramentas de ETL*, *aplicações do usuário* e *host seguro*; já o servidor é composto por *filas*, *processadores*, *memória* e *DaaS*. Cada um desses compo-

nentes é descrito a seguir. Por convenção, a sigla *RU* representa uma *Requisição do Usuário*, que pode ser inclusão de dados ou consultas analíticas, e *RC* denota uma *Requisição Criptografada*, a qual representa uma requisição do usuário transformada com base no esquema de um DWC correspondente.

Figura 4 – Arquitetura do sistema proposto



Fonte: O autor.

As *aplicações do usuário* são os programas de terceiros usados pelo usuário para emitir consultas analíticas. Este componente permite que consultas analíticas do usuário sejam definidas com base no esquema lógico de um DW, o qual é conhecido do usuário. Isso propicia a utilização transparente da criptografia, pois o usuário não necessita conhecer o esquema do DWC correspondente, tampouco ter ciência de que a criptografia está sendo aplicada. Assim, as aplicações do usuário enviam consultas analíticas do usuário para o *motor de execução* do *host seguro*, que inicia o fluxo de processamento das consultas sobre DWC armazenados no *DaaS* no servidor.

As *ferramentas de ETL* são aquelas fornecidas por terceiros e usadas para fazer a carga de dados. Como as *aplicações do usuário*, estas ferramentas conhecem somente o esquema lógico de um DW, logo enviam registros de dados baseados neste esquema ao *motor de execução* do *host seguro*, que inicia o fluxo de processamento da carga de dados num DWC mantido no *DaaS* no servidor.

O *módulo de registro* compreende as interfaces usadas por administradores para registrar as regras de mapeamento entre os esquemas lógicos de um DW e do DWC correspondente e as regras que definem quais e como as operações de consultas analíticas são mapeadas para serem executadas sobre um DWC. Este também é usado para obter as chaves de criptografia dos administradores, as quais são necessárias para criptografar nomes de tabelas e atributos de um DWC, itens de dados a serem incluídos num DWC e parâmetros definidos em consultas analíticas, e também para decodificar itens de dados de um DWC e os resultados de consultas.

O *host seguro* é responsável por realizar toda a responsabilidade do ambiente do usuário e inclui um conjunto de *metadados*, *motor de execução* e *módulo de criptografia*.

Os *metadados* são mantidos num repositório e equivalem aos esquemas lógicos dos DW e DWC correspondentes, às regras de mapeamento entre esquemas lógicos e de consultas analíticas e às chaves de criptografia.

O *motor de execução* é o componente responsável por efetivamente cadastrar esquemas, mapeamentos e chaves de criptografia no repositório de *metadados* e por coordenar o processamento de uma RU. Isso inclui as obrigações de mapear RU em RC, de enviar RC para execução no servidor, de obter os resultados de RC do servidor; de decodificar os resultados quando uma RU for uma consulta analítica; e de repassar os resultados para as *aplicações do usuário*. O *motor de execução* é acionado pelas *aplicações do usuário*, *ferramentas de ETL* e *módulo de registro*, de modo que os usuários possam acessar as funcionalidades do sistema, conforme descritas na Seção 4.1, e aciona o *módulo de criptografia* para executar as funções de criptografia e decodificação necessárias ao processamento de RU.

Cada RU é processada numa *thread* pelo *motor de execução*, de maneira que várias RU sejam executadas em paralelo pelo sistema. O *motor execução* também é a interface entre o cliente e o servidor na arquitetura: cada *thread* enfileira uma RC na *fila requisição* no servidor (resultante do mapeamento da RU solicitada) e espera uma resposta enviada pelo servidor por meio da *fila resposta*. Ao obter uma resposta, a *thread* recupera os resultados da RC que estão na *memória resultados* no servidor, para prosseguir com o processamento (i.e. decodificar os resultados e encaminhá-los às interfaces do usuário).

O *módulo de criptografia* contém as funções que efetivamente realizam os processos de criptografia e de decodificação. Estas funções são fornecidas pelos administradores, como bibliotecas de classes que são acionadas pelo *motor de execução*.

As *filas* são usadas para desacoplar a interação entre o *motor de execução* no cliente e os componentes do servidor bem como desacoplar a interação entre os componentes do próprio servidor. Além disso, sua filosofia *first in first out* permite o processa-

mento de RC por ordem de chegada. A *fila requisição* é o local usado pelo *motor de execução* para notificar algum *processador* alocado no servidor sobre uma nova RC a ser executada. A *fila processamento* é utilizada por algum *processador* do servidor para notificar outros *processadores* sobre a execução de uma RC em algum *DW-i* no *DaaS*, em que $i \in [1...m]$. A *fila status* é usada por algum *processador* para notificar o término da execução de uma RC em algum *DW-i* no *DaaS*. Já a *fila resposta* é usada por algum *processador* para notificar o *motor de execução* sobre o término do processamento de uma RC.

O *DaaS* é o local de armazenamento escalável dos dados de um DWC, os quais são particionados entre vários *DW-i*. O armazenamento escalável é um recurso fornecido nativamente por alguns provedores de DaaS, tal como o *SQL Azure Database*, do qual este sistema faz uso. Um *DW-i* é um SGBD fornecido por um provedor de DaaS que embute os mecanismos de armazenamento e de execução de consultas. Este componente, de fato, mantém os dados criptografados num banco de dados e processa as consultas analíticas sobre os dados criptografados. Além dos *DW-i*, o *DaaS* contém o componente *gerente de DW*, que é um repositório de metadados sobre os *DW-i*. Este é consultado pelos *processadores* do servidor para obter os endereços de *DW-i*.

A *memória* é o local de armazenamento dos resultados de uma RC. Ela é dividida em *memória temp*, que armazena os resultados parciais de um RC, quando é uma consulta analítica; e *memória resultados*, que armazena os resultados finais de uma RC. A *memória temp* é acessada e usada pelo conjunto de *processadores* do servidor, enquanto a *memória resultados* é usada pelos *processadores* e pelo *motor de execução* do *host seguro*.

Os *processadores* são *máquinas* alocadas para prover um mecanismo escalável de processamento de consultas analíticas sobre DWC mantidos no *DaaS* e para executar a inclusão de dados nos DWC. Quando um *processador* obtém uma RC na *fila requisição* que é uma inclusão de dados, o processador realiza as seguintes tarefas: obtém do *gerente de DW* no *DaaS* o endereço do *DW-i* em que o registro de dados deve ser incluído, acessa o *DW-i* e envia um comando de inclusão para ser processado no *DW-i*.

Quando a RC obtida é uma consulta analítica, o *processador* tem a função de *orquestrador* e, portanto, é responsável por coordenar a execução da RC. Isto inclui as seguintes tarefas: consultar o *gerente de DW* no *DaaS* para obter os endereços dos *DW-i*; distribuir a execução da RC entre os *processadores* por meio da fila *processamento*; recuperar o status da execução dos *processadores* por meio da fila *status*; recuperar os resultados parciais da execução da RC dos *processadores* na *memória temp*; unir os resultados parciais obtidos e gerar os resultados finais; registrar os resultados finais na *memória resultados*; e notificar o *motor de execução* sobre o término da execução da RC por meio da *fila resposta*.

Quando um *processador* obtém uma RC na *fila processamento*, este funciona como *escravo*, logo é responsável por: fazer algum pré-processamento na consulta analítica, requerido pelas técnicas de criptografia adotadas [p. ex. Lopes *et al.* (2014) requerem que consultas analíticas sejam reescritas para que o SGBD não compute agrupamentos de dados]; submeter a RC para processamento num *DW-i* no *DaaS*; realizar algum pós-processamento na consulta também requerido pelas técnicas de criptografia adotadas; armazenar os resultados dessa execução na *memória temp*; e notificar o *orquestrador* sobre o término do processamento usando a *fila status*.

Os *processadores* no servidor são executados assincronamente. Isso implica que, para uma dada RC, um *processador* pode ser o *orquestrador* da RC, enquanto que o mesmo *processador* pode ser um *escravo* de outra RC em execução no sistema. Além disso, os componentes *filas*, *processadores*, *memória* e *DaaS*, descritos na arquitetura proposta, são tipicamente serviços disponibilizados por provedores comerciais de computação em nuvem, tais como *Microsoft Azure* e *Amazon Web Services*.

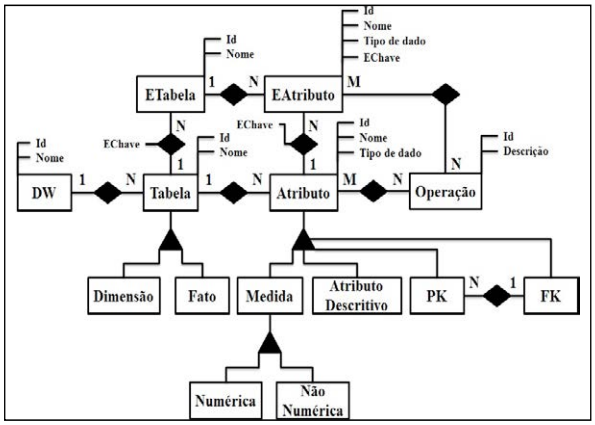
4.3 Metadados

A Figura 5 apresenta o esquema do repositório de metadados do sistema proposto. Uma instância da entidade *DW* se relaciona com muitas instâncias da entidade *Tabela*. Uma *Tabela* representa uma tabela de dimensão (especialização *Dimensão*) ou uma tabela de fatos (especialização *Fato*) e contém muitas instâncias da entidade *Atributo*, que é especializada em *Medida*, *Atributo Descritivo*, *PK* (chave primária) e *FK* (chave estrangeira). Já a entidade *Medida* é especializada em *Numérica* e *Não Numérica*. Instâncias da entidade *Atributo* podem se relacionar com muitas instâncias da entidade *Operação*, que representam as operações requeridas em consultas analíticas.

ETabela e *EAtributo* são, respectivamente, as entidades que representam as tabelas e atributos de

um DWC. Instâncias desta última se relacionam com múltiplas instâncias da entidade *Operação*, indicando quais operações de consultas analíticas são feitas sobre um atributo do DWC.

Figura 5 – Esquema do repositório de metadados



Fonte: O autor.

Quanto aos atributos, uma entidade *DW* contém *Id* e *Nome*. O mesmo para as entidades *Tabela* e *ETabela*, tal que o *Nome* de uma instância de *ETabela* é obtido aplicando-se a chave *EChave*. Uma entidade *Atributo* contém *Id*, *Nome* e *Tipo de Dado*, enquanto *EAtributo* contém *Id*, *Nome*, *Tipo de Dado* e *EChave*, que é a chave usada na criptografia dos valores do atributo. O nome de uma instância de *EAtributo* é obtido usando-se a chave *EChave*. Finalmente, uma instância de *Operação* é caracterizada pelos atributos *Id* e *Descrição*.

4.4 Módulo de criptografia

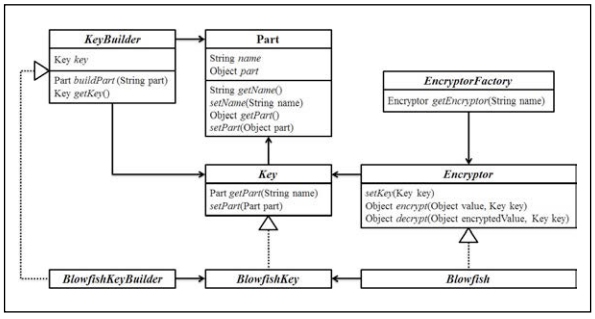
Este módulo é uma biblioteca de classes que encapsula as técnicas de criptografia usadas na encriptação dos nomes de tabelas e atributos de um DWC e na encriptação e decodificação de itens de dados e de parâmetros e resultados de consultas analíticas. Por padrão no sistema, nomes de tabelas e atributos de um DWC são criptografados usando *Blowfish* (SCHNEIER, 1994). Por sua vez, as técnicas de criptografia aplicadas na encriptação e decodificação de itens de dados e de parâmetros e resultados de consultas são especificadas pelos administradores do sistema.

A Figura 6 ilustra o diagrama de classes do *módulo de criptografia*. A interface *Encryptor* determina um contrato de uso para funções de criptografia e de decodificação. Por exemplo, a classe *Blowfish* realiza esta interface de tal forma que possa ser aplicada na encriptação dos nomes de tabelas e atributos de um DWC.

Uma chave de criptografia implementa a interface *Key*, a qual define um contrato para a obtenção das partes de uma chave. Por exemplo, a *BlowfishKey* implementa *Key* para o *Blowfish*. Já o contrato de construção de uma chave de criptografia é feito pela interface *KeyBuilder*. Por exemplo, *BlowfishKeyBuilder* é uma classe que realiza a interface *KeyBuilder* para construir uma chave *BlowfishKey*.

EncryptorFactory apresenta uma interface para a obtenção de uma instância de um *Encryptor* concreto a partir de seu nome. Sua implementação é baseada numa API de reflexão, em que é possível criar em tempo de execução uma classe concreta em função do nome fornecido.

Figura 6 – Classes do módulo de criptografia



Fonte: O autor.

4.5 Modelo de ameaças

O sistema proposto está sendo projetado para admitir um modelo de ameaças em que o servidor é considerado um ambiente vulnerável aos ataques de um adversário *honesto, porém curioso*. Admite-se que este adversário não tem permissões no cliente, mas possui privilégios de administrador no servidor. Isso inclui o acesso aos dados criptografados mantidos na *memória* e no *DaaS* e às mensagens das *filas*, assim como o acesso aos *logs* de execução de consultas dos *processadores* e dos SGBD no *DaaS*. O adversário deseja conhecer os dados criptografados a fim de tentar obter os dados não criptografados correspondentes. Ele não tem, contudo, a intenção de modificar a semântica das consultas analíticas e do processamento de consultas e nem de modificar os dados criptografados armazenados na *memória* e no *DaaS*.

O sistema proposto pode prevenir estas ameaças ao processar consultas analíticas sobre um DWC sem requerer a decodificação de dados no servidor. Além disso, as informações sobre o esquema lógico de um DWC não revelam a semântica do DW correspondente, pois nomes de tabelas e atributos de um DWC são criptografados no cliente, que é considerado seguro e inacessível pelo adversário. Também, o adversário não tem acesso às RU e aos seus mapeamentos para RC, em que as informações obtidas são apenas sobre a RC. As RC se referem aos itens de dados criptografados, às consultas analíticas baseadas no esquema lógico de um DWC e aos parâmetros e resultados de consultas criptografados. Desse modo, explorando as mensagens registradas nas *filas* e nos logs de execução dos *processadores* e dos SGBD no *DaaS*, o adversário é incapaz de descobrir informações sobre o esquema lógico do DW e sobre as chaves de criptografia, dados não criptografados e valores dos parâmetros definidos nas consultas do usuário, pois estas informações não são enviadas ao servidor.

Assim, o sistema pode proteger a confidencialidade de dados contra um adversário que não conhece o esquema lógico de um DW e as consultas analíticas formuladas pelo usuário bem como os mapeamentos de RU em RC, as chaves de criptografia e os dados não criptografados, pois tais informações são obtidas a partir do cliente, o qual é considerado inacessível pelo adversário.

4.6 Limitações do sistema

Quanto ao aspecto da confidencialidade de dados, o sistema revela as seguintes informações ao servidor: as RC em execução; as consultas analíticas processadas e em processamento num DWC; os dados criptografados armazenados no *DaaS* e na *memória*; as estruturas das tabelas de um DWC, incluindo os nomes de tabelas e atributos, o número de atributos, o número de registros e os tipos de dados dos atributos; e os relacionamentos entre as tabelas. Além disso, não define formas de gerenciar a revogação das chaves de criptografia e é incapaz de proteger a confidencialidade de dados quando um adversário obtém acesso ao cliente.

Quanto ao processamento escalável, o sistema não é capaz de se recuperar de falhas na execução de uma RC em particular, pois se o *orquestrador* de uma RC falhar, o sistema não prossegue com o processamento desta RC.

Quanto ao processamento de consultas analíticas sobre um DWC, o sistema é capaz de executar somente as operações sobre os dados criptografados que são possíveis com as técnicas de criptografia sendo aplicadas.

5 Conclusão e trabalhos futuros

Neste artigo foi especificado um sistema de processamento de consultas analíticas sobre DWC e armazenados na nuvem, o qual propõe funcionalidades inovadoras em relação ao estado da arte: é *extensível*, pois admite o uso de diferentes técnicas de criptografia sobre um DW; é *assíncrono*, pois os *processadores* executam requisições recuperadas de uma *fila*, sem aguardar o fim de uma requisição anterior; e é *escalável*, pois os *processadores* executam requisições em paralelo e independentes.

O sistema proposto é baseado em serviços disponibilizados por provedores de computação em nuvem, como *processadores*, *filas* e *DaaS*. Além disso, não requer mudanças nas arquiteturas dos SGDB providos pelo *DaaS*; permite executar consultas analíticas sobre um DWC sem decodificação de dados na nuvem; possibilita a escalabilidade dos dados de um DWC e a paralelização da execução de consultas analíticas sobre um DWC; e protege a confidencialidade de dados contra ataques de um adversário *honesto, porém curioso*, ao servidor.

Este sistema está sendo implementado com a pretensão de se investigar o impacto de diferentes técnicas de criptografia e também o impacto da escalabilidade no desempenho de consultas analíticas sobre DWC e mantidos na nuvem.

REFERÊNCIAS

ABADI, D. J. Data Management in the Cloud: Limitations and Opportunities. **IEEE Data Engineering Bulletin**. Washington, v. 21, n. 1, p. 3–12, 2009.

BABY, T.; CHERUKURI, A. K. On Query Execution Over Encrypted Data. **Security and Communication Networks**, New York, v. 8, p. 321-331, 2015.

BADGER, M. L. *et al.* Cloud Computing Synopsis and Recommendations. **National Institute of Standards and Technology**, 2012. Disponível em: <http://www.nist.gov/manuscript-publication-search.cfm?pub_id=911075>. Acesso em: 10 jun. 2015.

CHUNG, S. S.; OZSOYOGLU, G. Anti-Tamper Databases: Processing Aggregate Queries over Encrypted Databases. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING WORKSHOPS, 22., Atlanta. **Proceedings...** Atlanta: ICDEW, 2006, p. 98-106.

CODD, E. F.; CODD, B. S.; SALLEY, C. T. **Providing OLAP (Online Analytical Processing) to User-Analysts: An IT Mandate**. 1. ed. Codd & Associates, 1993.

GOSAIN, A.; ARORA, A. Security Issues in Data Warehouse: A Systematic Review. **Procedia Computer Science**, Atlanta, v. 48, p. 149–157, 2015.

HACIGUMUS, H. *et al.* Executing SQL over Encrypted Data in the Database-Service-Provider Model. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 22., Madison. **Proceedings...** Madison: SIGMOD, 2002, p. 216-227.

HACIGUMUS, H.; IYER, B.; MEHROTRA, S. Efficient Execution of Aggregation Queries over Encrypted Relational Databases. In: INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS, 9., Jeju Island. **Proceedings...** Jeju Island: DASFAA, 2004, p. 125-136.

HORE, B. *et al.* Secure Multidimensional Range Queries over Outsourced Data. **The VLDB Journal**, Secaucus, v. 21, n. 3, p. 333–358, 2012.

INMON, W. H. **Building the Data Warehouse**. 4. ed. New York: Wiley, 2005.

KADHEM, H.; AMAGASA, T.; KITAGAMA, H. A Novel Framework for Database Security based on Mixed Cryptography. In: INTERNATIONAL CONFERENCE ON INTERNET AND WEB APPLICATIONS AND SERVICES, 4., Venice. **Proceedings...** Venice: ICIW, 2009, p. 163-170.

KADHEM, H.; AMAGASA, T.; KITAGAMA, H. Optimization Techniques for Range Queries in the Multivalued-Partial Order Preserving Encryption Scheme. In: FRED, A.; DIETZ, J. L. G.; LIU, K.; FILIPE, J. (Org.). **Knowledge Discovery, Knowledge Engineering and Knowledge Management**, Springer Berlin Heidelberg, 2013, v. 272, p. 338-353.

KIM, J. *et al.* On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1. In: INTERNATIONAL CONFERENCE ON SECURITY AND CRYPTOGRAPHY FOR NETWORKS, 5., Maiori. **Proceedings...** Maiori: SCN, 2006, p. 242-256.

KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit**. 3. ed. New York: Wiley, 2013.

KOUNS, J.; MARTIN, B. DataLossDB: Open Security Foundation. **OSF DataLossDB**, 2015. Disponível em: <http://www.datalossdb.org>. Acesso em: 10 jun. 2015.

LIU, D. Securing Outsourced Databases in the Cloud. In: NEPAL, S.; PATHAN, M. (Org.). **Security, Privacy and Trust in Cloud Systems**. Springer Berlin Heidelberg, 2014, p. 259-282.

LOPES, C. C.; TIMES, V. C. A Framework for Investigating the Performance of Sum Aggregations over Encrypted Data Warehouses. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 30., Salamanca. **Proceedings...** Salamanca: ACMSAC, 2015, p. 1000-1007.

LOPES, C. C. *et al.* Processing OLAP Queries over an Encrypted Data Warehouse Stored in the Cloud. In: International Conference on Data Warehousing and Knowledge Discovery, 16., Munique. **Proceedings...** Munique:DaWaK, 2014, p. 195-207.

MELL, P.; GRANCE, T. The NIST Definition of Cloud Computing. **National Institute of Standards and Technology**, 2011. Disponível em: <http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909616>. Acesso em: 10 jun. 2015.

POPA, R A. **Building Practical Systems That Compute on Encrypted Data**. 154 f. 2014. Thesis (Doctoral Degree of Philosophy)–Department of Electrical Engineering and Computer Science, MIT, Massachusetts, 2014.

POPA, R. A.; LI, F. H.; ZELDOVICH, N. An Ideal-Security Protocol for Order-Preserving Encoding. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY, 2013., San Francisco. **Proceedings...** San Francisco: SSP, 2013, p. 463-477.

POPA, R. A. *et al.* CryptDB: Processing Queries on an Encrypted Database. **Communications of the ACM**, New York, v. 55, n. 9, p. 103-111, 2012.

SRINIVASAMURTHY, S. *et al.* Security and Privacy in Cloud Computing: A Survey. **Parallel & Cloud Computing**, New York, v. 2, n. 4, p. 126-149, 2013.

SANTOS, R. J.; BERNARDINO, J.; VIEIRA, M. A Survey on Data Security in Data Warehousing: Issues, Challenges and Opportunities. In: INTERNATIONAL CONFERENCE ON COMPUTER AS A TOOL, 2011, Lisboa. **Proceedings...** Lisboa: EUROCON, 2011, p. 1-4.

SANTOS, R. J. *et al*. A Specific Encryption Solution for Data Warehouses. In: MENG, W.; FENG, L.; BRESSAN, S.; WINIWARTER, W.; SONG, W. (Org.). **Database Systems for Advanced Applications**, Springer Berlin Heidelberg, v. 7826, p. 84-98, 2013.

SMITH, K. *et al*. Making Query Execution Over Encrypted Data Practical. In: JAJODIA, S.; KANT, K.; SAMARATI, P.; SINGHAL, A.; SWARUP, V.; WANG, C. (Org.). **Secure Cloud Computing**. Springer New York, 2014, p. 171-188.

SCHNEIER, B. Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). In: ANDERSON, S. (Org.). **Fast Software Encryption**. Springer Berlin Heidelberg, 1994, p. 191-204.

SOUSA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios. In: ESCOLA REGIONAL DE COMPUTAÇÃO CEARÁ, MARANHÃO E PIAUÍ, 3., **Anais...** ERCEMAPI, 2009, p. 150-175.

THOMPSON, B. *et al*. Privacy-Preserving Computation and Verification of Aggregate Queries on Outsourced Databases. In: PRIVACY ENHANCING TECHNOLOGIES SYMPOSIUM, 9., Seattle. **Proceedings...** Seattle:PETS, 2009, p. 185-201.

TU, S. *et al*. Processing Analytical Queries over Encrypted Data. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 39., Trento. **Proceedings...** Trento: VLDB, 2013, p. 289-300.

WU, Z. *et al*. Executing SQL Queries over Encrypted Character Strings in the Database-As-Service Model. **Knowledge-Based Systems**, Amsterdam, v. 35, p. 332-348, 2012.