Information Sciences 422 (2018) 497–515

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A symmetric cryptographic scheme for data integrity verification in cloud databases

Luca Ferretti[a,*], Mirco Marchetti[a], Mauro Andreolini[b], Michele Colajanni[a]

[a] University of Modena and Reggio Emilia, Department of Engineering "Enzo Ferrari", Modena, Italy
[b] University of Modena and Reggio Emilia, Department of Physics, Informatics and Mathematics, Modena, Italy

ARTICLE INFO

Article history:
Received 14 February 2017
Revised 8 September 2017
Accepted 10 September 2017
Available online 14 September 2017

Keywords:
Authentication
Integrity
Bloom filter
MAC
Cloud
Database

ABSTRACT

Cloud database services represent a great opportunity for companies and organizations in terms of management and cost savings. However, outsourcing private data to external providers leads to risks of confidentiality and integrity violations. We propose an original solution based on encrypted Bloom filters that addresses the latter problem by allowing a cloud service user to detect unauthorized modifications to his outsourced data. Moreover, we propose an original analytical model that can be used to minimize storage and network overhead depending on the database structure and workload. We assess the effectiveness of the proposal as well as its performance improvements with respect to existing solutions by evaluating storage and network costs through micro-benchmarks and the TPC-C workload standard.

1. Introduction

Cloud database services represent an important opportunity for many enterprises and organizations attracted by high availability and scalability guarantees. However, their adoption is limited by the perceived risks about data confidentiality and integrity that can be violated by insiders and external attackers [32]. In this paper we consider the problem of data integrity in cloud databases, and we propose an efficient solution to assess the integrity of outsourced data while minimizing network and storage overhead.

We should consider that cloud service contracts do not oblige providers to notify tenants about data corruption, hence verifying data integrity remains a tenant's burden. Existing verification solutions are affected by prohibitive computational, storage and bandwidth overhead that have an impact on costs because additional network and storage usage increases cloud service expenses [22]. We propose a novel solution for integrity verification that allows a tenant to efficiently detect unauthorized modifications on outsourced data at reduced storage and network overhead. The contribution of this paper is threefold:

- it defines a novel integrity protocol and analyzes the guarantees that it offers;
- it proposes an analytical model that allows the tenant to optimize the parameters of the proposed integrity protocol with the goal of minimizing storage and network overhead as a function of database workload characteristics;

* Corresponding author
E-mail addresses: luca.ferretti@unimore.it (L. Ferretti), mirco.marchetti@unimore.it (M. Marchetti), mauro.andreolini@unimore.it (M. Andreolini), michele.colajanni@unimore.it (M. Colajanni).

http://dx.doi.org/10.1016/j.ins.2017.09.033
0020-0255/

- it evaluates the performance of the proposed protocol through micro-benchmarks and the TPC-C standard database benchmark.

The literature on database outsourcing faces three types of correctness guarantees: *integrity, completeness* and *freshness* [10,30,46,50]. Completeness and integrity are satisfied if the result of a query includes all and only the relevant data that an authorized party inserted in the database, thus guaranteeing that tenant's data is not altered, deleted or ignored by the outsourced database. Freshness ensures that a client receives the latest version of the requested data. This paper focuses on integrity and proposes a novel scheme that allows efficient detection of unauthorized modifications of data stored in cloud databases.

Existing solutions to guarantee data integrity leverage cryptographic digests that can be based on asymmetric or symmetric primitives. Asymmetric cryptographic accumulators [7,39] ensure optimal asymptotic complexity in storage, computation, and network usage [35], but their excessive computational costs prevent their adoption in most database scenarios [15]. Symmetric Message Authentication Codes (MAC) can guarantee the integrity of tenant files stored in the cloud [2], but their adoption in cloud database services poses several challenges related to the granularity of the protected data: if every value is protected by a MAC, the database storage increases considerably; if the entire set of rows/tables are protected by one MAC, then the verification of each value requires to fetch an entire row/table, thus imposing an excessive network overhead that renders verification unfeasible.

The proposed scheme relies on a variant of Bloom filters [8] for the detection of unauthorized modifications to outsourced data. This choice guarantees two benefits: the processes of integrity verification and update of the authentication structures cause low network and storage overhead; all cryptographic operations are based on symmetric schemes that do not introduce significant computation overhead. The performance of the proposed scheme depends on Bloom filter sizes that should be chosen on the basis of the integrity guarantees required by the tenant. We enrich the proposal by offering an analytical model that takes as its input the integrity requirement, the database characteristics and workload, and evaluates the optimal size of the Bloom filters that minimize overheads and related cloud service costs.

Our proposal is orthogonal to data encryption strategies for data confidentiality proposed in literature [20,21,23,43], and can be integrated with encryption algorithms to achieve both confidentiality and integrity of data stored in cloud databases. Moreover, it can be used to design solutions that aim to ensure data completeness and freshness [47].

To the best of our knowledge, the proposed scheme is the most efficient solution for detecting unauthorized modifications in cloud database services. We demonstrate its benefits through micro-benchmarks and the standard TPC-C benchmark. All results show that the proposed scheme greatly reduces network and storage footprint with respect to existing proposals.

The remaining part of the paper is organized as follows. Section 2 describes the considered cloud database scenario and the threat model. Section 3 outlines theoretical background on Bloom filters. Section 4 presents the proposed solution and its security guarantees. Section 5 describes how the tenant must size the protocol parameters to achieve the required security level. Section 6 presents the analytical method to minimize overhead. Section 7 discusses performance in terms of storage and network overhead, and compare results against state-of-the-art schemes. Section 8 compares our solution with existing proposals. Finally, Section 9 concludes the paper by summarizing its main contributions and future work. For further implementation details, security analyses and proofs please refer to the Appendices.

2. Scenario and threat model

We consider a scenario where a *tenant* stores large amounts of data into a cloud database through clients that issue read and write operations. The tenant benefits from pay-per-use cloud prices with the goal of reducing his operational costs, but his data at rest, in motion and in use are subject to different security threats.

Our proposal aims to improve security of data in use against internal malicious attackers, and can be combined with additional solutions to protect other attack surfaces. For example, clients and database servers should adopt the SSL/TLS protocol suite to protect confidentiality and integrity of data in motion as well as the ability to detect replay and reflection attacks. The cloud provider must own a valid PKI certificate that avoids man-in-the-middle attacks. All clients must own valid credentials (e.g., API tokens, client-side PKI certificates) that allow the provider to identify and authenticate them, as well as grant proper access on the resources stored in the database.

In the proposed scheme, we assume that all clients share the same secret key, that can be distributed according to known key distribution schemes [9], as well through more efficient strategies that are specific to the field of cloud database services [16,20]. We assume that the secret key is not known by the cloud provider nor by any other part that is not authorized to manipulate tenant data. We assume that all clients are trusted and will never send corrupted data to the cloud database, nor will leak confidential information to unauthorized parties (including the cloud provider). On the other hand, we assume that the cloud provider is not trusted and could alter tenant's data. Modification may be caused by hardware or software failures, as well as by deliberate attacks coming from external adversaries or from insiders within the cloud organization. From the tenant's point of view, any unauthorized modification represents a data integrity violation.

Data integrity and authenticity is a prominent research area of the cryptography community, and is usually guaranteed by means of message authentication codes (MAC) [5]. A MAC applied to arbitrary data together with a secret key produces a *(cryptographic) digest* (also called *tag*), that is a compressed representation of the input data. An attacker can violate integrity by forging a digest on behalf of the authorized users. The security level of a MAC depends on the key and digest