Available online at www.sciencedirect.com

ScienceDirect

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

An evaluation of recent secure deduplication proposals

Vladimir Rabotka, Mohammad Mannan *

Concordia Institute of Information Systems Engineering, Concordia University, Montreal, QC, Canada

ARTICLE INFO

Article history:
Available online 26 October 2015

Keywords:
Cloud storage
Privacy
Deduplication

ABSTRACT

Deduplication is widely used by cloud storage providers to cut costs, by storing and uploading a single instance of identical files shared across multiple user accounts. However, cross-account deduplication introduces several new side-channel attacks on user privacy; see e.g., Harnik et al. (IEEE Security and Privacy Magazine, 2010), Mulazzani et al. (USENIX Security, 2011). As a response, several solutions have been proposed to mitigate different deduplication privacy concerns. In this paper, we summarize notable attacks on deduplication, and analyze recently proposed privacy-preserving secure deduplication solutions in terms of privacy-gain, deployment and bandwidth costs, and security limitations (if any). In particular, we identify weaknesses in a secure deduplication proposal based on the use of a home gateway device (Heen et al., New Technologies, Mobility and Security, 2012); we also explore how these weaknesses may lead to three separate attacks. Overall, our analysis may help storage providers to evaluate competing solutions, and the research community to better design privacy-preserving deduplication solutions by addressing limitations of current proposals.

1. Introduction

Popular cloud storage services boast users in the millions, with gigabytes of free storage offered to each user. To leverage common files shared across user accounts, several cloud storage services use data deduplication. Deduplication eliminates the need to upload and store redundant copies of user data, by verifying before each upload if a file (or, more generally, a data block) already exists in the server's storage. If so, the file is not uploaded and the corresponding user account is simply linked to the existing file on the server. Data deduplication is believed to save significant storage and bandwidth costs.

For example, a recent empirical study (Meyer and Bolosky, 2011) on 857 desktop computers reports that with deduplication, the storage requirement is only about 32% of the original storage size (see also Harnik et al., 2012 for an efficient estimation of deduplication ratios).

Serious privacy concerns may arise when deduplication is used by popular storage services. Harnik et al. (2010) explore several side-channel attacks; for example, the presence of a specific file in the cloud can be verified (by observing network traffic), and linked to a specific user, e.g., by having access to a file that uniquely identifies a target user. An attacker can also fill out a template file with specific details of a target victim (e.g., salary in an employment contract or diagnosis in a medical record template), and infer the existence of such a file in the cloud. Mulazzani et al. (2011) demonstrate several attacks on Dropbox due to the use of deduplication (see also Soghoian, 2011). For example, an attacker can obtain access to an existing file, simply by supplying the hash of the file (i.e., without possessing the content of the file); the original owner of the

* Corresponding author. Concordia Institute of Information Systems Engineering, Concordia University, Montreal, QC, Canada. Tel.: 514-848-2424 ext. 8972; fax: 514-848-3171.

E-mail address: m.mannan@concordia.ca (M. Mannan).

http://dx.doi.org/10.1016/j.jisa.2015.08.001

file remains oblivious to the attack. Halevi et al. (2011) provide example scenarios in which hash values of sensitive files may be exposed and the file content accessed by attackers.

To counter known attacks, several academic solutions have been proposed over the past few years. Most proposals can be placed into one of four categories: (i) encryption-based solutions; (ii) probabilistic uploads; (iii) proof-of-ownership schemes; and (iv) gateway-based solutions. In this paper, we analyze representative solutions from each category in terms of their effectiveness (i.e., privacy-gain), deployment and operational costs, and security weaknesses (if any).

Client-side encryption of user data seems to be an obvious solution to several deduplication attacks. However, a straightforward use of encryption can eliminate advantages of deduplication and incur a high penalty in storage and bandwidth consumption for cloud providers (e.g., the same file will generate a different ciphertext for each user who uploads it). Several proposed solutions (e.g. Douceur et al., 2002; Sheng et al., 2011; Liu et al., 2015) aim to provide confidentiality against a storage provider, and still allow for deduplication.

Probabilistic upload-based solutions (e.g. Harnik et al., 2010; Halevi et al., 2011; Lee and Choi, 2012) attempt to improve user privacy by requiring additional uploads (e.g., randomly requesting uploads for an already uploaded file). The primary goal is to confuse an attacker about whether a target file exists in the cloud or not. However, these solutions do not offer strong deployment incentives for service providers, as they significantly increase bandwidth costs due to the extra uploads.

Using proof of ownership (PoW) (Cachin and Schunter, 2011) schemes, a server can verify that a user is in full possession of a file without the need of a full upload, before linking the server copy to the user's account. Several PoW-based schemes have been proposed in the recent past, focusing primarily on efficiency gains (e.g. Zheng and Xu, 2012; Shin et al., 2012b; Halevi et al., 2011; Xu et al., 2013; Fan et al., 2012; Di Pietro and Sorniotti, 2012). Other deduplication goals, such as proof of data possession (PoD) (Ateniese et al., 2007; Wang et al., 2011), and proof of retrievability (PoR) (Bowers et al., 2009; Juels and Kaliski, 2007; Shacham and Waters, 2008), have also been proposed. We do not address PoD and PoR solutions here, since we consider scenarios of curious-but-honest cloud storage providers.

Heen et al. (2012) propose a deployment-friendly, home gateway-based solution to address privacy attacks due to deduplication. It is assumed that the user's network service provider (NSP) is also the cloud storage provider. The home gateway device is deployed by the NSP. As stated (Heen et al., 2012), in some countries (e.g., UK, France), NSPs already offer cloud storage services, which may favor home gateway-based solutions. Our analysis of Heen et al.'s proposal identifies several potential weaknesses that may be exploited to launch known side-channel attacks. We also attempt to fix these weaknesses, and analyze our proposed counter-measures.

The remainder of this paper is organized as follows. In Section 2 we provide the necessary background on deduplication and review known deduplication attacks. After analyzing current mitigation attempts based on encryption (Section 3), probabilistic uploads (Section 4) and proof of ownership schemes (Section 5), we discuss the home gateway-based solution proposed by Heen et al. (Section 6), presenting three attacks as well as possible ways of mitigating the attacks. A storage-gateway solution (Shin and Kim, 2015), which offers differential privacy is analyzed in Section 8. Existing solutions are compared in Section 9 in terms of security, deployment costs and target environments.

2. Background and known attacks

In this section, we briefly introduce the concept of deduplication, and discuss currently known attacks exploiting deduplication as used by cloud storage providers (CSPs). We later use these attacks to evaluate different privacy-enhanced deduplication proposals.

2.1. *Deduplication*

Deduplication eliminates duplicate copies of redundant data from a CSP. Data is stored and/or transferred only once. Subsequent copies are replaced by pointers to the one physical data instance. Deduplication approaches vary according to organizational needs. For instance, in *server-side deduplication*, data is always uploaded from the client to the CSP, but only one copy is stored on the server. This approach saves storage space but not bandwidth. In *client-side deduplication*, when a client wishes to upload a file to the cloud, a unique representation of the file (e.g., cryptographic hash) is sent to the storage provider. This unique representation is much smaller than the file itself and acts as a fingerprint for the file. If the file is already present in the cloud (e.g., identical hash), the file is linked to the client's account without performing an actual upload. This approach saves both storage space and network bandwidth.

By observing the amount of upload traffic and comparing it to the file size, an attacker can learn if a file was deduplicated (e.g., only the hash was transferred) or not (a full upload was performed). An attacker will measure the traffic between the CSP and her own machine and thereby identify if a particular file is present or absent in the cloud. This fact can be exploited when client-side deduplication is performed over different user accounts, and the physical copy of a file is shared across different and otherwise unrelated accounts.[1] Side-channel attacks on deduplication require only a valid account with the same CSP as the victim. No further traffic analysis is necessary (e.g., encrypting data in transit does not prevent or mitigate deduplication attacks).

2.2. *Side channel attacks on deduplication*

Harnik et al. (2010) propose three side channel attacks on user privacy by exploiting cross-user deduplication. We summarize the attacks below.

1. A single file can uniquely identify a user, exposing the user's identity. For instance, an organization can set up a trap, where different versions of a sensitive document are made

[1] Note that, for privacy reasons, not all CSPs use cross-user data deduplication; see, e.g., SpiderOak (Fairless, 2010) (but see also Wilson and Ateniese, 2014).