Using Cryptography to Control Your Data at a Distance

by Colin Boyd, Gareth T. Davies, Kristian Gjøsteen (NTNU), Håvard Raddum and Mohsen Toorani (University of Bergen)

Most people and companies store important information using cloud storage services that are outside their direct control. The information may be personal, such as emails, photos and videos, medical records and financial information. How can we be sure that our data is safe from the prying eyes of cloud operators, other cloud users or outside agencies? How can we be sure that our data will remain available to us when we need it?

Security of information is an essential aspect of business and government activity, whether it relates to protection of corporate knowledge, integrity of financial transactions, or reliable storage and transmission of data. The transition to cloud computing necessitates extra security measures to protect valuable data that is no longer under the direct control of its owner. This issue has been widely recognized; the industry-led Cloud Security Alliance (www.cloudsecurityalliance.org) was formed in 2008, and the NIST guidelines on cloud security and privacy were published in 2011 [2]. The Snowden revelations of 2013 and 2014 have changed the IT security priorities and it is now understood that there is an urgent need for protection of personal, business, and government data against pervasive monitoring and infiltration.

The collaborative project Cryptographic Tools for Cloud Security, funded by the Norwegian Research Council from 2016 to 2019, will study new cryptographic tools to enable cloud security against powerful attackers. The research involves experts at Norwegian University of Science and Technology and University of Bergen, in cooperation with University of Mannheim.

The new cryptographic primitives, protocol and models that we will develop will lead to theoretical advances as well as practical outcomes. We are following the current important trend in cryptographic research to connect rigorous results strongly to real-world usage. This is now both possible and timely given that a level of maturity has been reached in cloud computing which will allow us to demonstrate the practical effectiveness of our proposals, to complement the theoretical analysis.

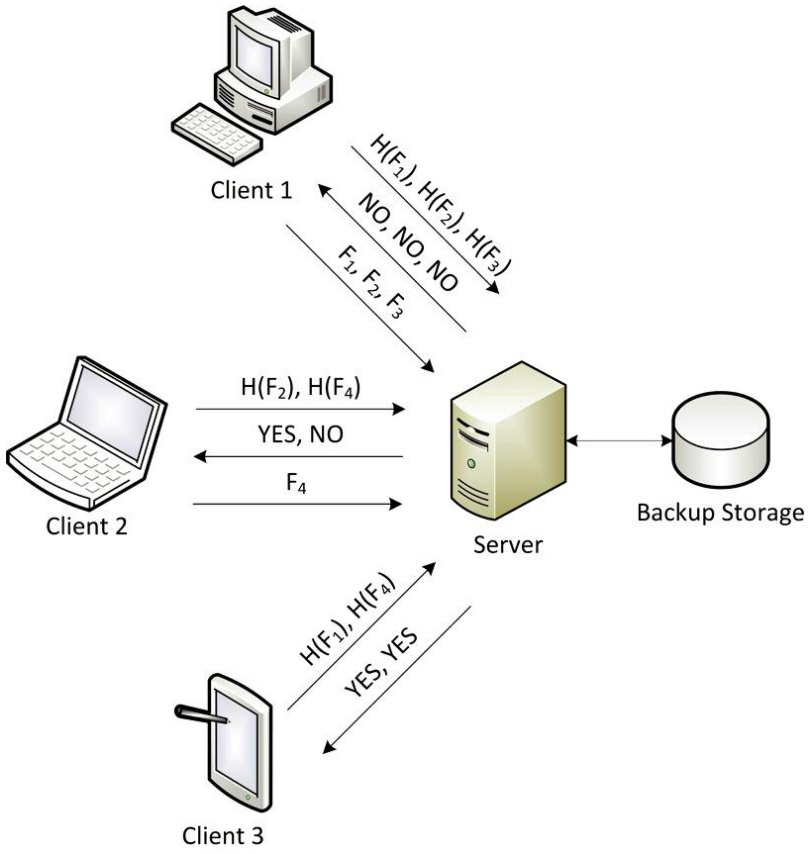

Figure 1: Simple client-side deduplication in which different clients sequentially request the server to store different files F_i. The client first sends hashes of the files, $H(F_i)$. The server checks if files with those hash values are already stored and, if not, the client sends the files.

Cryptography has traditionally been used to protect data while it is being transmitted over insecure networks or while it is at rest in static storage. These services remain important in the cloud as it is essential to protect both confidentiality and integrity of data while it is transmitted between client and cloud server and while it is at rest in cloud storage. At the same time, new approaches are also required for at least two reasons.

- Data in cloud storage is frequently shared between multiple parties, may be stored in geographically distributed nodes, and needs to be updated incrementally. This requires the development of practical techniques to allow cloud users to efficiently verify the integrity and availability of their data, including where it is located.
- We often want to process data in the cloud without necessarily trusting the cloud operator. Therefore, we want to be able to compute on encrypted data. This can include basic operations such as searching through encrypted records through to full-scale processing of any function. Gentry's theoretical breakthrough of fully homomorphic encryption in 2009 remains impractical in general, but exploring compromises which are both efficient and secure is an important theme of our project.

One main area of focus of our research is secure deduplication. By deduplication we mean that the server stores only a single copy of each file, regardless of how many clients asked to store that file, in order to make significant savings in both storage and bandwidth. Note that large files such as movies and software are very likely to be shared by many users. Generally, deduplication can be at client-side (which saves both storage and bandwidth) or server-side (which only saves storage). However, deduplication contrasts with users' desire for security: if two users A and B upload the same file encrypted under independent keys kA and kB, the server will receive independent ciphertexts and will thus be unable to perform deduplication.

One possible solution is to derive the encryption key from the file itself [1]; but this approach will only give security against an adversarial server with the unrealistic assumption that files are unpredictable. Other security issues arise irrespective of any encryption. For example, suppose that the cloud service provider (CSP) employs client-side deduplication (see Figure 1) in which the client first sends a short identifier to the CSP and the CSP tells the client to upload the full file only if it is not already stored. An adversarial user can create a template of a file (e.g., an employment contract of Bob) and attempt a number of uploads of files that only differ in one detail (e.g., salary) and at some point the upload will be halted by the CSP, meaning that this file is already stored (and thus learns Bob's salary) [3]. We are working on schemes which defend against such attacks by differentiating between files that are popular (and thus promise significant savings from deduplication) and those that are not.

References:

[1] J. R. Douceur et al.: "Reclaiming space from duplicate files in a serverless distributed file system", in IEEE Distributed Computing Systems, 2002, pp 617–624. IEEE, 2002.

[2] T. Grance, W. Jansen: "Guidelines on Security and Privacy in Public Cloud Computing", Special Publication 800-144, National Institute of Standards and Technology, December 2011.

[3] D. Harnik, B. Pinkas, A. Shulman-Peleg: "Side channels in cloud services: Deduplication in cloud storage", IEEE Security & Privacy, 8(6):40–47, 2010.

Please contact:

Colin Boyd, NTNU, Norway

colin.boyd@item.ntnu.no