TALLINN UNIVERSITY OF TECHNOLOGY
Department of Computer Science
TUT Center for Digital Forensics and Cyber Security

ITC70LT

Gvantsa Grigolia 144965

EVALUATION OF DATA OWNERSHIP SOLUTIONS IN REMOTE STORAGE.

Master Thesis

Supervisor: Ahto Buldas
Professor

Tallinn 2016

Autorideklaratsioon

Autorideklaratsioon on iga lõputöö kohustuslik osa, mis järgneb tiitellehele. Autorideklaratsioon esitatakse järgmise tekstina:

Olen koostanud antud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud. Käsolevat tööd ei ole varem esitatud kaitsmisele kusagil mujal.

Autor: [Ees− ja perenimi]

[June 23, 2016]

Annotatsioon

Annotatsioon on lõputöö kohustuslik osa, mis annab lugejale ülevaate töö eesmärkidest, olulisematest käsitletud probleemidest ning tähtsamatest tulemustest ja järeldustest. Annotatsioon on töö lühitutvustus, mis ei selgita ega põhjenda midagi, küll aga kajastab piisavalt töö sisu. Inglisekeelset annotatsiooni nimetatakse Abstract, venekeelset aga

Sõltuvalt töö põhikeelest, esitatakse töös järgmised annotatsioonid:

- kui töö põhikeel on eesti keel, siis esitatakse annotatsioon eesti keeles mahuga $\frac{1}{2}$ A4 lehekülge ja annotatsioon *Abstract* inglise keeles mahuga vähemalt 1 A4 lehekülg;
- kui töö põhikeel on inglise keel, siis esitatakse annotatsioon (Abstract) inglise keeles mahuga $\frac{1}{2}$ A4 lehekülge ja annotatsioon eesti keeles mahuga vähemalt 1 A4 lehekülg;

Annotatsiooni viimane lõik on kohustuslik ja omab järgmist sõnastust:

Lõputöö on kirjutatud [mis keeles] keeles ning sisaldab teksti [lehekülgede arv] leheküljel, [peatükkide arv] peatükki, [jooniste arv] joonist, [tabelite arv] tabelit.

Abstract

Võõrkeelse annotatsiooni koostamise ja vormistamise tingimused on esitatud eestikeelse annotatsiooni juures.

The thesis is in [language] and contains [pages] pages of text, [chapters] chapters, [figures] figures, [tables] tables.

Contents

List of Figures

List of Tables

1. Introduction

Describes the problem statement, illustrates why this is a problem and describes the contribution the thesis makes in solving this problem. Optionally, it can give a short description (1-3 sentences each) of the remaining chapters. Good introductions are concise, typically no longer than 4 pages.

2. Terms and definition

Defines the fundamental concepts your thesis builds on. Your thesis implements a new type of parser generator and uses the term non-terminal symbol a lot? Here is where you define what you mean by it. The key to this chapter is to keep it very, very short. Whenever you can, don't reinvent a description for an established concept, but reference a text book or paper instead.

3. Background and Related Work

3.1. Data Deduplication

Cloud computing is an on-demand service. Customers are charged based on used storage and bandwidth.[1] Both service providers and customers are interested in cost efficient solutions of cloud storage. Data deduplication offers disk and bandwidth savings. Idea is simple – avoid or remove a duplicated data. This section covers basic concepts of deduplication technology. Lists various methods and processing type sand underlines the approaches used in cloud storage.

3.1.1. Hash Based Deduplication

To remove or avoid duplicated data, it must be detected first.Hash based data deduplication uses the hash values of a file (or data chunk) as a file (or data chunk) identifier. Hashes of files are calculated and then are kept on the server. When the file is uploaded first time, its hash is computed and it is compared with the existing hashes on the server.If there is a match, the file is not stored on the disk (or in case of client-side deduplication, is not transfered at all).Instead, server creates the reference, which points on the already existing file, with the same hash value.If computed hash does not match with any of the hashes, the file together with the hash value is stored on the server.[1]

3.1.2. Types of Deduplication

Data deduplication differs based on processing methods. If it takes place before the client application transfers the file to the server,it is known as client-side deduplication. If it takes place, after the file is uploaded on server, it is known as server side deduplication. In client-side deduplication scenario, the client application computes the hash of the file and sends it

[1] "With Amazon S3, you pay only for the storage you actually use. There is no minimum fee and no setup cost. Amazon S3 has three pricing components: storage (per GB per month), data transfer in or out (per GB per month), and requests (per n thousand requests per month)." http://aws.amazon.com/s3/pricing/

to the server. If the hash already exists on the server side, client application does not send the file. If no match is found, it means,that the file is unique and client application transfers it to the server. On the other hand, if client application directly sends file to the server and server computes the hash after it, it is called server-side deduplication. Both processing methods save storage, but client-side deduplication also reduces bandwidth consumption.[2]

Apart of divers processing methods,data deduplication differs in processing levels. There are file and block level data deduplication. Difference between them is intuitive. In case of file level, hash of file is calculated and as a result unique files are stored on the server. In block level scenario , files are divided into blocks(fixed or variable size). Hashes of these blocks are calculated and duplicated data on block level is avoided. [2]

The last concept is, single and cross client data deduplication. Single client data deduplication removes duplicated data in scope of one user. Duplicated data will be stored on the server, if it belongs to different users. On the other hand, cross client deduplication vanishes the user boundaries and unique data would be shared among the users.[2]

3.1.3. Summary

Cloud storage providers are looking for, the most efficient way to reduce the cost.In cross user client-side deduplication case, file or ”chunks” of file are stored only ones on the disk and users are sharing the data. It reduces the bandwidth cost dramatically, because the deduplication takes palace on client side, and duplicated files are not uploaded at all.[3] Such cost reductions is attractive for cloud storage providers, but this technology has some security drawbacks. Next section covers potential attacks taking place during cross-user client-side data deduplication in cloud storage.

3.2. Confidentiality and Privacy Issues in Remote Storage

Although deduplication is a beneficial technology, there are security drawbacks,leading to potential attacks. Taking into consideration the behavior of the cross-user client-side data deduplication, it is easy to learn some general facts. This section focuses on attacks breaching the confidentiality of data and privacy of remote storage clients, when cross-user client-side data deduplication takes place.

3.2.1. Potential Attacks

Danny Harnik was first who has demonstrated, the potential attacks in remote storage related to data deduplication technology.[4] The paper covers three cases: file detection, file content detection and covert channel. The first case shows, how trivial is to learn whether the remote server already contains the particular file or not. Attacker uploads the file and observes the network traffic or the time required to upload the file. If the file already is stored on the server, there is no need to upload it again. Client application sends only the hash of the file to the server. The observer detects, that amount of data is smaller then file's size itself.(Size of hash depends on hash function and is smaller then file size.) If file is "big enough", observing the time required to upload file on server, is sufficient to learn the fact. The law enforcement authorities, can use this behavior. Check if storage provider contains the file (e.g. file's content is against the law) and later, they can force remote storage service providers to revile the identity of the file owner.

Data deduplication technology opens the possibility to guess the content of user's data.The approach is straightforward, attacker just tries all possible variations and waits for occurrence of data deduplication. Once it takes place, attacker learns that such file (file with this content) exist on the server. The trick is that, unlike the dictionary attacks it is not detectable. It is the legitimate way to upload new documents on the server.[4] This type of attack is easy to lunch against the files with small min-entropy. To have batter understanding, refer to the following example. Bob is invited at the event in the cinema. He stores his invitation ticket in the cloud. Alice wants to learn the row and the place of Bob's ticket. She put the Bob's name on the right place and starts to brute force row and place numbers. Alice generates files with different content and uploads on cloud. Once the deduplication takes place, she will get the desired information.

Last case describes the covert channel. Precondition for this scenario is, that attacker already have to own the victims machine. In order to exchange one bit information "0" or "1", attacker generates two random files and uploads one of them. If the first file is uploaded the covert channel transfers "0" else it transfers "1" bit. Covert channel can transfers more information, by altering the number of files or the meaning of file.[4]

Later Halevi address the main weakness of deduplication.[5] He states that, to use hash as a proxy to retrieve the file is vulnerable. Owning a small static piece of the file(e.g. hash of the file) does not necessarily mean owning the entire file. He introduces the Proof of Ownership Protocol, which dramatically reduces the probability of the attacker to retrieve the file, without owning it. Next section[**?**] covers the detail description of Proof of Ownership Protocol and other solutions offered to substitute the hash as a proxy approach for data ownership in remote storage.

3.3. Summary

The amount of savings offered by data deduplication, depends on data type and content produced by users of such services.[6] In case of office workers as users in remote storage, the benefit from deduplication is high. Office workers use mostly identical template to generate the data and the portion of duplication is high. Applying data deduplication technology saves bandwidth and disk space. But same time it rises privacy and confidentiality issues. The major weakness is that, client-side deduplication is detectable and using hash as a proof of ownership is not sufficient. Anyone who possesses the hash value of file, is able to retrieve the file from the server. If the attacker obtains the hash of the file, he can retrieve the file from the server and gain unauthorized access to it.

Data privacy issue in cloud computing is one of the aspects that could break the trust of the users towards the service providers. So those who what to stay on the market, should build the systems, which takes into consideration privacy and confidentiality issues.

4. Approach

- Proofs of Ownership in Remote Storage Systems - 2011
- Boosting Efficiency and Security in Proof of Ownership for Deduplication - 2012
- Provable Ownership of File in De-duplication Cloud Storage - 2013
- A Secure Client Side Deduplication Scheme in Cloud Storage Environments - 2014
- A Tunable Proof of Ownership Scheme for Deduplication Using Bloom Filters - 2014
- An efficient confidentiality-preserving Proof of Ownership for Deduplication -2014

5. Evaluation

Demonstrate why the developed framework is secure and efficient.

1. Time complexity evaluation.
2. Cost analyses.

6. Future Work

In science folklore, the merit of a research question is compounded by the number of interesting follow-up research questions it raises. So to show the merit of the problem you worked on, you list these questions here.

7. Conclusions

Short summary of the contribution and its implications. The goal is to drive home the result of your thesis. Do not repeat all the stuff you have written in other parts of the thesis in detail. Again, limit this chapter to very few pages. The shorter, the easier it is to keep consistent with the parts it summarizes.

References

[1] M. B. Babette Haeusser, Alessio Bagnaresi and A. Woodcock, *Guide to Data De-duplication:The IBM System Storage TS7650G ProtecTIER De-duplication Gateway*. IBM Redbooks, 2008, pp. 20–27.

[2] S. Srinivasan, *Security, Trust, and Regulatory Aspects of Cloud Computing in Business Environments*. USA: Information Science Reference (an imprint of IGI Globalo), 2014, pp. 77–78.

[3] M. H. T. K. M. R. U. V. Moritz Borgmann, Tobias Hahn and S. Vowe, *On the Security of Cloud Storage Services*. Germany: SIT Technical Reports, 2012, pp. 55–110.

[4] B. P. Danny Harnik and A. Shulman-Peleg, "Side channels in cloud services, the case of deduplication in cloud storage," *IEEE Security and Privacy Magazine*, vol. 8, no. 2, pp. 40–47, 2010.

[5] B. P. Shai Halevi, Danny Harnik and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," *ACM Conference on Computer and Communications Security*, pp. 491–500, 2011.

[6] M. Dutch, "Understanding data deduplication ratios," pp. 5–9, 2009.

A. Appendix 1