RESeED: A Tool for Regular Expression Search over Encrypted Data in Cloud Storage

Mohsen Amini Salehi[1], Thomas Caldwell, Alejandro Fernandez, Emmanuel Mickiewicz, Eric W. D. Rozier[2], and Saman Zonouz[3]
Electrical and Computer Engineering, University of Miami
{*m.aminisalehi*[1], *e.rozier*[2], *s.zonouz*[3]}*@miami.edu*

David Redberg
Fortinet Inc.
Sunnyvale, California 94086
dredberg@fortinet.com

***Abstract*—We present RESeED, a tool that provides user-transparent and Cloud-agnostic regular expression search over encrypted data without requiring trust in the Cloud, or changes to Cloud infrastructure. Upon receiving a search query, RESeED translates it to a finite automata and analyzes efficient and secure representations of the data before asking the Cloud to download the matching encrypted files. We demonstrate and evaluate a working prototype of RESeED and show the scalability and correctness of our approach using data from `arXiv.org`.**

I. Introduction

Cloud providers offer scalable storage solutions to users and relieve the burden and costs of managing a data center. In spite of the advantages provided by the Cloud services, there is increasing concern over the confidentiality of user data stored within the Cloud.

A proven solution to confidentiality concerns is the use of cryptographic techniques for user data [2]. However, such techniques limit search capabilities on data stored in the Cloud. The establishment of algorithms which allow for keyword searches over encrypted documents [1] has been critical to the development of privacy preserving search in Cloud environments. However, one powerful tool which has remained elusive is the ability to apply regular-expression based search on encrypted data. A solution using current methods, such as PEKS [1], remains infeasible in practice due to the exponential explosion of the space required for the storage of the resulting ciphertexts.

In this demo, we present *RESeED*, a tool that provides a scalable and Cloud-agnostic algorithm for regular expression search over encrypted files stored in the Cloud. RESeED achieves this objective without cooperation from the Cloud provider or the need for special infrastructure. RESeED uses local indexing and symmetric encryption to provide deployment efficiency and information leakage guarantees. RESeED enables users to upload files to the Cloud for storage, remotely search for the encrypted contents of the files using regular expressions, and download the data that matches these queries. It minimizes end-user involvement by implementing the necessary encryption and decryption steps in the background.

This work demonstrates the following features:

- A novel and scalable solution for searching regular-expressions over encrypted data in the Cloud. Our solution operates based on two novel data structures as well as algorithms that process search queries using these data structures.

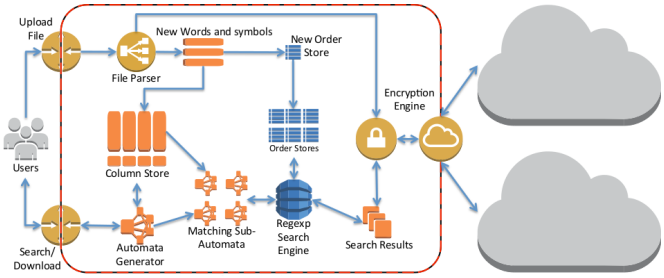

Figure 1. RESeEDArchitecture

- Efficacy of RESeED when compared with existing solutions that are used to search unencrypted data.

II. Design and Implementation

The architecture of RESeED is illustrated in Figure 1. Search queries are processed through the use of novel data structures that we call the *column store* and *order store*. The column store is used to index keywords found in each file in the data-set. The order store, which contains a fuzzy representation of keyword ordering, for each file through the use of fuzzy hashes.

Using these data structures, our proposed algorithms can process search queries given in the form of a regular expression by users. First, a search query is converted into a non-deterministic finite automaton (NFA). This automaton is then partitioned into a set of sub-NFAs based on the appearances of delimiters in the search expression. In the next step, the algorithm checks for the presence of each token of the column store in each sub-NFA. After finding the set of files that match all sub-NFAs, the algorithm uses the order store for these files and uses a path-NFA generated based on the matched sub-NFAs to confirm the keywords appear in such a way as to form an accepting path in the original NFA. If we find a match, the encrypted file is marked as part of the set of files which contain a match for the original regular expression.

RESeED is deployed on trusted hardware that facilitates secure access to Cloud providers. Currently, RESeED is built to be deployed on Fortivault, a trusted gateway for accessing Cloud services provided by Fortinet Pty. Ltd.[1]

[1]http://www.fortinet.com/

III. Demo Description

Although RESeED has been designed to be executed on trusted hardware, for demonstration purposes, we have made it accessible at the following web address: http://www.performalumni.org/trust/Dragonfruit/demo/.

This demo includes the following steps:

- **Upload File**: Uploads a file to Cloud. While RESeED is designed to be Cloud-agnostic. For this demo, we use Dropbox[2] as the Cloud storage. The current version of RESeED accepts the `pdf` and `txt` file formats.
- **Creating order store**: A user creates an order store with an arbitrary hash-width (*i.e.,* size of the hashed tokens in the order store). The hash-width impacts the imposed overhead and false positive rate of RESeED. Our evaluations showed that a hash-width of three leads to low overhead and an almost zero false positive rate.
- **Search**: Users can search contents of the files uploaded to the Cloud using regular expressions formatted with the syntax shown in Table I.
- **Downloading the search results**: RESeED displays a list of files matching any search. These files can then be downloaded and decrypted.

Table I
Regular expression symbols and their meanings.

Symbol	Definition
.	any character; Character . is shown as \.
*	zero or more repetition of a character
+	one or more repetition of a character
?	zero or one repetition of a character
|	OR statement
\s	any separator character
\w	any alphabetic character
\d	any numeric character

In order to evaluate the performance and correctness of RESeED, we tested it on a collection of scientific papers from the `arXiv.org`[3] repository. This data-set contains 683,620 `pdf` files with the total size of 264 GB. All experiments were conducted on a computer running Linux (Ubuntu 12.04), with an Intel Xeon processor (1.80 GHz) and 64 GB RAM. We use a set of seven regular expression benchmarks that are listed in Figure 2. They are sorted based on the execution time of RESeED for these searches.

(A) All files that have "Cloud Computing" in their text:
`cloud(\s)+computing`
(B) Structured Query Language or SQL:
`S(tructured)?(\s)+Q(uery)?(\s)+L(anguage)?`
(C) All references to TCP/IP or Transmission Control Protocol Internet Protocol:
`((Transmission(\s)*Control(\s)*Protocol)|(TCP))(\s)*/?(\s)*((Internet(\s)*Protocol)|(IP))`
(D) All dates with YYYY/MM/DD format:
`(19|20)(\d\d)/(0(1|2|3|4|5|6|7|8|9)|1(0|1|2))/(0(1|2|3|4|5|6|7|8|9)|(1|2)\d|3(0|1))`
(E) URLs that include Computer Science (cs) or Electrical and Computer Engineering (ece) and finished by .edu:
`http://((\w|\d)+\.)*(cs|ece)\.(\w|\d|\.)+\.edu`
(F) All IEEE conference papers after the year 2000:
`(2\d\d\d(\s)+IEEE(\s)+(\w|\s)*)|(IEEE(\s)+(\w|\s)*2\d\d\d(\s))Conference`
(G) Any XML scripts in the papers:
`<(\?)?(\s)*(xml|html)(\s)+.*(\?)?>`

Figure 2. Regular expression benchmarks used for evaluations.

[2]http://www.dropbox.com/

[3]http://arxiv.org/

IV. Evaluation of Benchmarks

We evaluated RESeED's performance on the `arXiv.org` data-set and compared it's performance against the `grep` utility[4]. For each benchmark, we measured the overall search time for RESeED, indicating the time to construct automata, the time to match against the column store, and the time to match against the order store. We also measured the total time that `grep` takes to search the same regular expression over the **unencrypted** data-set.

Figure 3 shows the result of our evaluations using the benchmarks listed in Figure 2. The experiment shows the feasibility of searching complicated regular expressions within a limited time. The figure shows that even though our method searches on the encrypted data, it executes faster for benchmarks (A)-(D) when compared to `grep`. The reason for this speed up is that our method uses the column store to identify files which could contain a match, searching fewer files compared to `grep` which scans the whole file set. We note that for the benchmarks that take longer to execute than `grep` ((E)-(G)), our method spends a considerable amount of its time processing the column store. In general, our method outperforms `grep` when given less fuzzy regular expressions or when the list of the order stores that need to be searched is small. In the former case, matching each entry of the column store against our generated automata is performed quickly and in the latter case, the number of files that have to be checked in the order store are few.

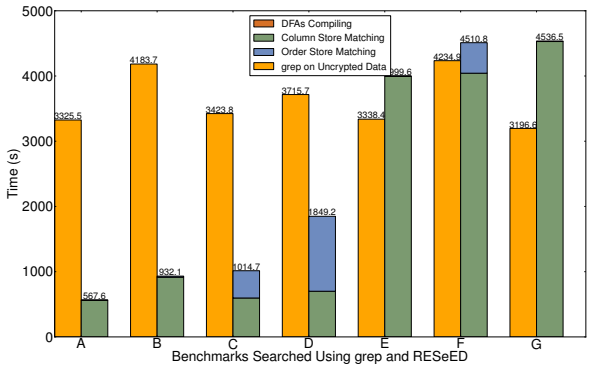

Figure 3. Times to search benchmarks in Figure 2 for `grep` and RESeED.

V. Conclusion

In this paper, we have presented RESeED, a method which provides Cloud providers with a user-transparent and Cloud-agnostic capability to process regular expression based search over encrypted data residing in the Cloud. Our experiments on a real-world data set show RESeED's deployability and practicality empirically.

References

[1] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology-Eurocrypt 2004*, pages 506–522, 2004.

[2] E. W. D. Rozier, S. Zonouz, and D. Redberg. Dragonfruit: Cloud provider-agnostic trustworthy cloud data storage and remote processing. In *Proc. of the 19th IEEE Pacific Rim Int. Symp. on Dependable Computing*, PRDC '13, 2013.

[4]http://pubs.opengroup.org/onlinepubs/9699919799/utilities/grep.html

APPENDIX

More algorithms and experiments on the contributions of this study can be found in the Technical Report available in the following address:

http://www.performalumni.org/trust/Dragonfruit/techreport/