資訊安全 Final Project

5105056013 吳嘉偉、5105056029 江青霞、5105056019 廖健智

2017/06/25

1 Introduction

Many online services let users query large public datasets: some examples include restaurant sites, product catalogs, stock quotes, and searching for directions on maps. In these services, any user can query the data, and the datasets themselves are not sensitive. However, web services can infer a great deal of identifiable and sensitive user information from these queries, such as her current location, political affiliation, sexual orientation, income, etc.

2 Proposed Method

2.1 Splinter

2.1.1 Architecture

There are two main principals in Splinter: the user and the providers. Each provider hosts a copy of the data. Providers can retrieve this data from a public repository or mirror site. For a given user query, all the providers have to run it on the same view of the data. A user splits her query into shares, using the Splinter client, and submits each share to a different provider. The user can select any providers of her choice that host the dataset. The providers use their shares to execute the user's query over the cleartext public data, using the Splinter provider library. As long as one provider is honest (does not collude with others), the user's sensitive information in the original query remains private. When the user receives the responses from the providers, she combines them to obtain the final answer to her original query.

2.1.2 Security Goals

The goal of Splinter is to hide sensitive parameters in a user's query. Specifically, Splinter lets users run parametrized queries, where both the parameters and query results are hidden from providers.

Splinter supports a subset of the SQL language, hides the information represented by the questions marks and the query' s re-

sults, but the column names being selected and filtered are not hidden.

3 Case Studies

4 Related Work

5 Experimental Result

6 Conclusion

2.1.3 Threat Model

Splinter keeps the parameters in the user' s query hidden as long as at least one of the user-chosen providers does not collude with others. Splinter also assumes these providers are honest but curious: a provider can observe the inter- actions between itself and the client, but Splinter does not protect against providers returning incorrect results or maliciously modifying the dataset.

It assume that the user communicates with each provider through a secure channel (e.g., using SSL), and that the user' s Splinter client is uncompromised. The cryptographic assumptions are standard. They only assume the existence of one-way functions in our two-provider implementation. In our implementation for multiple providers, the security of Paillier encryption is also assumed.

7 Reference

[1] Splinter: Practical Private Queries on Public Data Frank Wang, Catherine Yun, Shafi Goldwasser, Vinod Vaikuntanathan, Matei Zaharia† MIT CSAIL, †Stanford InfoLab

[2] E. Boyle, N. Gilboa, and Y. Ishai. Function secret sharing. In Proceedings of the 34th Annual International Confer- ence on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), pages 337-367. Sofia, Bul- garia, Apr. 2015.

[3] N. Gilboa and Y. Ishai. Distributed point functions and their applications. In Proceedings of the 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), pages 640 - 658. Copenhagen, Denmark, May 2014.