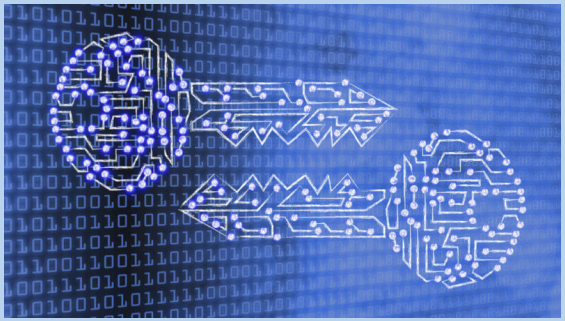
Using FAME for Auditing in a Data-sharing Network

by the Puppy Hackers

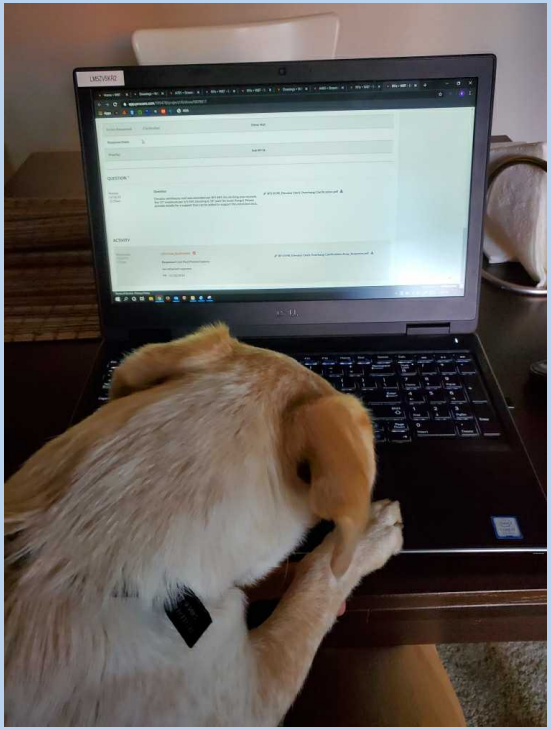
Mollie Zechlin from Spring Labs

January 15, 2021



Overview

1. Auditing in a Data-sharing Network
2. Common Solutions
3. Why Functional Encryption?
4. Description of Project
5. Live Demo
6. Questions and Contact Info



Spring

Auditing in a Data-sharing Network

- Alice and Bob want to share data with each other through a network controlled by a 3rd party server
- They want to encrypt the data to restrict access from the 3rd party (or another person who hacks the database)
- They want to enable auditing without leaking information to this 3rd party

Spring

Common solutions

- All participants get the same secret key and this secret key is given to the auditor when necessary
  - Problem: Management of secret key

Common solutions

- All participants get the same secret key and this secret key is given to the auditor when necessary
  - Problem: Management of secret key
- Second copy of encryption for Auditor
  - Problem: Duplication of storage

Common solutions

- All participants get the same secret key and this secret key is given to the auditor when necessary
  - Problem: Management of secret key
- Second copy of encryption for Auditor
  - Problem: Duplication of storage
- Managed by manual policies
  - Problem: Clear target for compromise

Common solutions

- All participants get the same secret key and this secret key is given to the auditor when necessary
  - Problem: Management of secret key
- Second copy of encryption for Auditor
  - Problem: Duplication of storage
- Managed by manual policies
  - Problem: Clear target for compromise

Spring

Why Functional Encryption?

- Using Attribute-based Encryption we can enable Participants on the network to encrypt data for different groups of people (Auditor and other Participants) without compromising security or storage.

- This can also enable Participants to encrypt for further subsets of Participants by generating further policies.

Spring

Description of Project

- The participants are able to use attributes to manage who is able to decrypt their ciphertext by implementing policies.
- I focused on an auditing case in which a participant may enable auditing by encrypting it with the policy "Participant OR Auditor" and therefore either another participant or an auditor (as designated by the managing server) would be able to decrypt the ciphertext.
- This method also enables the participants who distinguish between types of participants who would be able to decrypt their ciphertext.

Live Demo

What does the Live Demo do?

- Initiates an ABE FAME scheme with 5 different policies and 6 different roles (with 4 possible attributes)
- One participant encrypt 5 different plaintexts with 5 different policies
- Each of the 6 given roles attempt to decrypt all 5 ciphertexts

Further Developments

Although it is not displayed in the demo, this project also focused on layering this encryption in a fashion that would allow for a transmission of one ciphertext that had another ciphertext revealed once decrypted allowing only a further subset of attribute-owners to see the second layer ciphertext.

Questions and Contact Info

Questions?

Questions and Contact Info

Questions?

Contact Info: Mollie Zechlin
mollie@springlabs.com

Spring