CSCI 2390 Final Project Proposal
GDPR Compliant CryptDB Proxy

Motivation

CryptDB provides a way to ensure user data protection when facing a curious/malicious DBA or an attacker who gains control over the DBMS server. In this final project, the curious DBA threat model will be extended to a profit-maximizing company that will use all data on hand to generate more revenue. This project aims to explore whether GDPR, especially user consent and right of erasure clauses, can be enforced by applying CryptDB to an application.

CryptDB

Policy annotation will play an essential part in this project. Currently, policy annotations are defined upon creation of schema. In other words, the responsibilities of defining policies fall on the shoulders of programmers. In this project, we will assume programmers are willing to and capable of implementing GDPR compliant policies. However, the user of the webapp should be able to modify privacy settings whenever they want. Thus, a mechanism to change policy annotation should be in place. Specifically, the proxy needs to translate the user's request for updating privacy setting (eg. user chooses to opt out of using his/her data for direct marketing) to a SQL query that removes the access_key of the respective principal that speaks for the user. Deletion of user information upon user request should also be in place for a GDPR compliant CryptDB. The mechanism involves user specifying which columns to delete. CryptDB's proxy should be able to translate this request to SQL queries to delete respective rows as well as any speak_for relationship.

Onions should also be under consideration. For example, a user in a Slack group can choose to opt out of direct marketing service but still wants to receive messages from the group. Essentially all onions regarding any type of *join* should be modified such that the service provider cannot aggregate this user's data into its marketing algorithm. All other onion layers should not be modified to preserve functionality of the main Slack service.

Approach

Application should allow user to modify policies and proxy should be in charge of the translation of policy into SQL queries. Thus, I will try to build a simple application and modify CryptDB's proxy to evaluate whether CryptDB is a good choice for enforcing GDPR's consent and deletion clauses.

The evaluation will be done through a series of tests and mock attacks. Tests will ensure that policy change and deletion of data will take place no matter what the user's and administrator's login statuses are. Mock attacks will explore if the data are still encrypted upon two situations: proxy being compromised and application being compromised. Furthermore, metrics on the timeliness of deletion and policy changes will be collected and discussed.