6.857 Final Project Proposal
Range Queries and Bulk Inserts in CryptDB

Hrishikesh Joshi, Max Kolysh, Mari Miyachi, John Wang

Introduction

Our group plans to work on a secure system for storing data within CryptDB. We are interested in adding support for new operations on encrypted data, and in particular, geographic range queries. The implication of this is that database administrators can only access encrypted data, so even if the system is compromised, the data cannot be decrypted. We will focus on implementing an order-preserving, homomorphic encryption scheme, which allows for mathematical operations on encrypted data. Our ultimate goal is to implement a series of partially homomorphic encryption schemes that will support geographic range queries within CryptDB.

Homomorphic Encryption

An important application of cryptography is to securely store data. Indeed, almost all large companies today require at least some amount of encryption for their data. With the rising amount of information being spread on the internet comes a corresponding need to keep particular pieces of information private (such as credit card or social security numbers). Encrypting every piece of information entering a database is impractical, however, because of the computational load required to decrypt, perform operations, then encrypt the data again.

Homomorphic encryption aims to solve this problem by operating on the encrypted data directly. This type of encryption has direct practical applications, such as the construction of secure databases. Homomorphic encryption allows a mathematical operation on encrypted data which leads to modified ciphertext. This new ciphertext can be decrypted into plaintext which matches the operation being performed on the original plaintext.

Many homomorphic encryption schemes have been proposed in the literature. For instance, RSA allows multiplication on ciphertext.[1] Multiplying two encrypted RSA ciphertexts is equivalent to multiplying two plaintexts raised to some power *x*. Since this is equivalent to taking the product

[1] R. L. Rivest, A. Shamir, and L. M. Adleman. (1978). *A Method For Obtaining Digital Signatures and Public-Key Cryptosystems.* Communications ACM, 21(2):120-126.

of the two plaintexts, then raising to the power *x,* one can decrypt the resulting multiplied ciphertext to produce multiplied plaintext.

Other encryption schemes, such as the Goldwasser-Micali cryptosytem, allow multiplication on ciphertexts as well.[2] The multiplication of two encrypted plaintexts in the Goldwasser-Micali scheme results in the addition modulo two of the underlying plaintexts.

What's more, fully homomorphic encryption schemes have been proposed. These systems provide an encryption schemes which can perform both addition and multiplication operations on ciphertext. Craig Gentry (2009) was the first to create such a system using cryptographic lattices.[3] Using Gentry's scheme, one can perform arbitrary computations on the ciphertexts. Since Gentry's seminal work in 2009, much research has attempted to create a practical implementation of the fully homomorphic scheme. Gentry and Halevi (2010) mades strides to implement a fully homomorphic encryption scheme. It's running time on computations, however, remains prohibitively expensive.[4] Unfortunately, no fully homomorphic encryption scheme has been made practical enough for commercial use.

To address this issue, we propose to help develop a practical cryptography system which implements many partially homomorphic encryption schemes. The mixture of these schemes provides an aggregate scheme which can implement a large number of frequently used operations in a database.

CryptDB

CryptDB is a system developed by CSAIL that allows an online application to execute database queries on encrypted data. This prevents an adversary from gaining access to an application's unencrypted data, even if the database is compromised.

CryptDB uses three techniques to achieve security: an *SQL-aware encryption strategy*, *adjustable query-based encryption*, and *chaining encryption keys to user passwords.* We focus on the first and second techniques in particular, as our project will involve augmenting the cryptosystem to allow for new database operations.

CryptDB allows the database to perform most SQL operations (selects, adds, creates, etc.) on cyphertext instead of plaintext by using data encryption schemes specifically catered to these queries. The system takes advantage of the idea that these operations are made up of a

[2] S. Goldwasser, S. Micali (1984). *Probabilistic Encryption*. Journal of Computer and System Sciences 28(2): 270–299.

[3] Gentry, C. 2009. *Computing Arbitrary Functions of Encrypted Data.* Association for Computing Machinery. http://crypto.stanford.edu/craig/easy-fhe.pdf

[4] Gentry, C. and Halevi, S. *A Working Implementation of Fully Homomorphic Encryption.* http://eurocrypt2010rump.cr.yp.to/9854ad3cab48983f7c2c5a2258e27717.pdf

“well-defined set of primitive operators, such as equality checks, order comparisons, aggregates (sums), and joins”.[5] Additionally, to expose only the minimal amount of data required for a particular query from the DBMS, CryptDB uses layers of encryption on each of the database entries. A cyphertext can thus be decrypted “just enough” to expose the homomorphism required to perform a particular query.

Proposal

Our goal is to augment the functionality of CryptDB to allow for geographic range queries. Specifically, given the ciphertext of two (latitude, longitude) coordinates, we want to calculate the distance between two coordinates. Also, we would like to add bulk inserts and updates to CryptDB. This would allow a user to more efficiently insert data, since currently, CryptDB will encrypt each item separately. By allowing for bulk inserts, only a single database connection will be needed, reducing the number of round trips required between user and database.

First, our team will need to fully understand how CryptDB accomplishes its security goals. Specifically, we will need to understand in detail the cryptosystems used for those queries already implemented in CryptDB. Our group shall coordinate with Raluca Ada Popa to get a better understanding of the current structure of the CryptDB codebase. We will look at and how the CryptDB system switches between encryption schemes, which will hopefully provide us with enough knowledge to begin implementing geographic range queries.

Next, we will develop our own cryptosystem that supports the necessary geo-queries. This encoding scheme will likely be a combination of the order-preserving encryption (OPE) and homomorphic encryption (HOM) techniques, as described by Ada Popa, Redfield et al.[6] This cryptosystem will be a standalone unit which can only be used for geographic queries.

Lastly, we will determine how our cryptosystem fits into the existing layers of adjustable query-based encryption, and we will implement the system in CryptDB.

Our group will conduct weekly meetings to divide work and evaluate our progress. We anticipate that the scope and scale of this project will be well within our means given the time frame and resources at hand.

[5] Ada Popa, R., Redfield, C., et al. *CryptDB: Protecting Confidentiality with Encrypted Query Processing*.http://people.csail.mit.edu/nickolai/papers/raluca-cryptdb.pdf
[6] Ibid.