CS @ ILLINOIS

Encrypting Databases to Mitigate Server Breaches

Abhiram Kothapalli, Rajgopal, Roshan Rajan, Samuel Lou
Mentor: Vincent Bindschaedler

Abstract

Our group created a novel protocol that allowed users and system admins to access encrypted data on a server using a dual key system. Server data encryption is done with a key that is partially split between a client and the server. Using this schematic, data can only be viewed and modified when the client is logged in and chooses to provide the key.

Motivation

Services like Google, Netflix, and Wordpress store a variety of sensitive client information such as emails, age, phone numbers, and passwords. Less secure databases are prone to hackers and malicious system administrators. Simply encrypting the database is not secure if the key is stored locally on the server or an adjacent database because it is easily accessible through a compromised server. Our group sought to create an improved encryption scheme that mitigates this risk.

Solution

Our encryption protocol consists of splitting the decryption key partially between the server and the client. Under this scheme, even if the database is breached, the adversary cannot decrypt information without the client half of the key. We have also designed a four tier encryption scheme to appropriately encrypt information with varying levels of security. For example user information such as age may require both the client and server key whereas user email may only require server side encryption. This allows certain data to be modified when the client is not connected.

Implementation

Our protocol was implemented upon the open source Wordpress server side codebase. All code modifications were written in PHP and were tested on a server provided by the Siebel Center Security Lab.

- Designed to encrypt the user email, password, activation key, and login information.
- Capable of key generation, and symmetric encryption using Defuse Security's PHP encryption library.
- Onion layered encryption using combination of full server key and full client key.

Results

- Gained a working familiarity with Wordpress codebase and PHP
- Gained a working understanding of security protocols used on networks
- Learned how to use cryptographic tools like onion encryption and symmetric/asymmetric keys

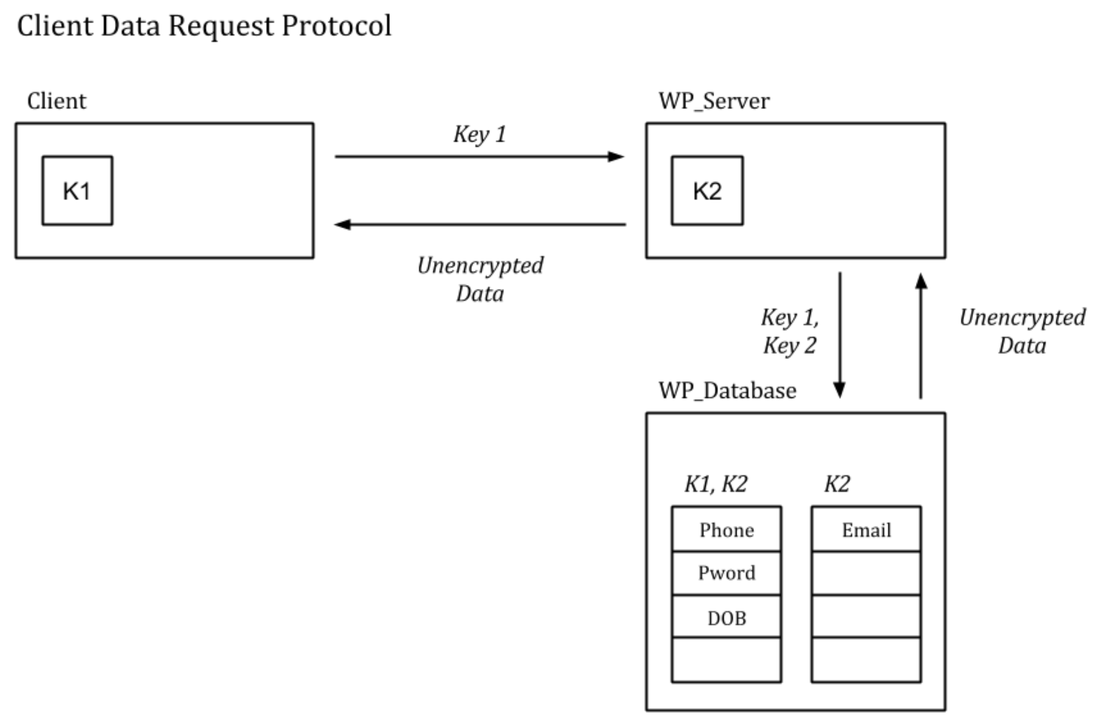

Figure 1: Protocol outlining a client data request. Client sends Key 1 which is merged with Key 2 and sent to the Database which returns the unencrypted data

P.U.R.E.
Promoting Undergraduate Research in Engineering