Secure HPTS requires Secure Hardware

Mr. X Et Al[1]

Microsoft Research

Several vendors including Amazon RDS and SQL Azure offer database-as-a-service functionality. Most organizations are making a significant bet on moving their database applications to the cloud. However several applications have sensitive information (e.g., employee salaries, customer PII) which needs to be secured in the cloud.

While most commercial database products support encryption, the solutions offered are often insufficient – they rely on encrypting the entire database file so that a disk-stealing adversary cannot obtain any sensitive information. But encrypted data is decrypted and stored as plaintext in memory and is accessible to a cloud admin. So, the only feasible solution in such cases is to encrypt the data before storing it in the cloud.

Consider a transactional workload which needs to do some non-trivial computation on encrypted data – for instance, a version of the debit-credit benchmark where each account balance is updated by a random percentage of its current balance. If the account balance column is encrypted using non-deterministic encryption (e.g., using AES in CBC mode), then the only way to compute the new balance is to ship the encrypted value to a secure client which has access to the keys, decrypt the value, compute the new value, re-encrypt it and store the resulting value in the server. This approach has two critical limitations: a) The DBMS is essentially used as a blob store which invalidates the use of stored procedures which are an important optimization to improve transaction processing performance by reducing communication between client and server b) The space overheads induced by encryption (encrypted values are usually 16 bytes) can be non-trivial – which can lead to large storage costs in the cloud. Thus, the TPS obtained at a particular price point can significantly decrease when we are dealing with transactions that require computation on encrypted data.

In this paper, we take the position that in order to facilitate efficient transaction processing on encrypted data we need to augment database servers *with secure hardware* in the cloud. The secure hardware enables the secure storage of encryption keys and can facilitate *computation on encrypted data in the cloud* without revealing the plaintext to an adversary. As a result, the database can execute full stored procedures independent of their complexity on encrypted data.

The talk will focus on the research challenges in building Cipherbase by modifying Microsoft SQL Server – a database system with a tightly coupled integration with secure hardware including a) Secure hardware design using FPGAs b) Optimization techniques to limit use of secure hardware c) Multi-row encryption techniques to reduce storage overheads d) Discussion of results of running popular transactional benchmarks on encrypted data.

---

[1] Mr. X denotes one or more of the project members of the Cipherbase project. http://research.microsoft.com/dbencryption/