TrustedDB: A Trusted Hardware based Database with Privacy and Data Confidentiality

Sumeet Bajaj
Stony Brook University
sbajaj@cs.stonybrook.edu

Radu Sion
Stony Brook University
sion@cs.stonybrook.edu

***Abstract*—TrustedDB is an outsourced database prototype that allows clients to execute SQL queries with privacy and under regulatory compliance constraints without having to trust the service provider. TrustedDB achieves this by leveraging server-hosted tamper-proof trusted hardware in critical query processing stages.**

TrustedDB does not limit the query expressiveness of supported queries. And, despite the cost overhead and performance limitations of trusted hardware, the costs per query are orders of magnitude lower than any (existing or) potential future software-only mechanisms. TrustedDB is built and runs on actual hardware, and its performance and costs are evaluated here.

I. Overview

Outsourcing has finally arrived, due in no small part to the availability of cheap high speed networks, storage and CPUs. Clients can now minimize their management overheads and virtually eliminate infrastructure costs

Virtually all major "cloud" providers today offer a database service of some kind as part of their overall solution. Numerous startups also feature more targeted data management and/or database platforms.

Yet, significant challenges lie in the path of large-scale adoption. Such services often require their customers to inherently trust the provider with full access to the outsourced datasets. But numerous instances of illicit insider behavior or data leaks have left clients reluctant to place sensitive data under the control of a remote, third-party provider, without practical assurances of *privacy* and *confidentiality* – especially in business, healthcare and government frameworks. And today's privacy guarantees of such services are at best declarative and subject customers to unreasonable fine-print clauses – e.g., allowing the server operator (or malicious attackers gaining access to its systems) to use customer behavior and content for commercial, profiling, or governmental surveillance purposes [5, 6].

Existing research addresses several such outsourcing security aspects, including access privacy, searches on encrypted data, range queries, and aggregate queries. To achieve privacy, in most of these efforts data is encrypted before outsourcing. Once encrypted however, inherent limitations in the types of primitive operations that can be performed on encrypted data lead to fundamental expressiveness and practicality constraints.

Recent theoretical cryptography results provide hope by proving the existence of universal homomorphisms, i.e., encryption mechanisms that allow computation of arbitrary functions without decrypting the inputs [12]. Unfortunately actual instances of such mechanisms seem to be decades away from being practical [7].

Ideas have also been proposed to leverage tamper-proof hardware to privately process data server-side, ranging from smartcard deployment [9] in healthcare, to more general database operations [3, 8, 10].

Yet, common wisdom so far has been that trusted hardware is generally impractical due to its performance limitations and higher acquisition costs. As a result, with very few exceptions [9], these efforts have stopped short of proposing or building full - fledged database processing engines.

However, *recent insights* [4] *into the cost-performance trade-off seem to suggest that things stand somewhat differently*. Specifically, at scale, in outsourced contexts, *computation inside secure processors is orders of magnitude cheaper than any equivalent cryptographic operation performed on the provider's unsecured common server hardware*[1], despite the overall greater acquisition cost of secure hardware.

This is so because cryptographic overheads (for cryptography that allows some processing by the server) are extremely high even for simple operations, a fact rooted not in cipher implementation inefficiencies but rather in fundamental cryptographic hardness assumptions and constructs (such as trapdoor functions – the cheapest we have so far being at least as expensive as modular multiplication [11]). This is unlikely to change anytime soon (none of the current primitives have, in the past half-century). New mathematical hardness problems (e.g., elliptic curve cryptography – which unfortunately is only a bit more efficient) will need to be discovered to allow hope of more efficient cryptography.

As a result, we posit that a full-fledged strong-privacy enabling secure database leveraging server-side trusted hardware can be built and run at a fraction of the cost of

[1] e.g., it is much cheaper to add numbers privately inside expensive cryptographic coprocessors than to perform the expensive cryptography needed to add the numbers encrypted on plain server hardware with privacy.

any (existing or future) cryptography-enabled private data processing on common hardware. We validate by designing and implementing TrustedDB, an SQL database processing engine that makes use of IBM 4764/5 [1, 2] cryptographic coprocessors programmed to run custom components securely.

Tamper resistant designs however are significantly constrained in both computational ability and memory capacity which makes implementing fully featured database solutions using secure coprocessors (SCPUs) very challenging. TrustedDB achieves this by utilizing common unsecured server resources to the maximum extent possible. For example TrustedDB enables the SCPU to transparently access external storage while preserving data confidentiality with on-the-fly encryption. This eliminates the limitations on the size of databases that can be supported. Moreover, client queries are pre-processed to identify sensitive components to be run inside the SCPU. Non-sensitive operations are offloaded to the untrusted host server. This greatly improves performance and reduces the cost of transactions.

Overall, despite the overheads and performance limitations of trusted hardware, the costs of running TrustedDB are orders of magnitude lower than any (existing or) potential future cryptography-only mechanisms. The TrustedDB design provides strong data confidentiality assurances. Moreover, it does not limit query expressiveness.

The contributions of this work are two-fold: (i) the introduction of new cost models and insights that explain and quantify the advantages of deploying trusted hardware for data processing, and (ii) the design, development, and evaluation of TrustedDB, a trusted hardware based relational database with full data confidentiality.

II. Conclusions

This work's inherent thesis is that, at scale, in outsourced contexts, computation inside secure hardware processors is orders of magnitude cheaper than any equivalent cryptography performed on a provider's unsecured common server hardware, despite the overall greater acquisition cost of secure hardware. We thus propose to make trusted hardware a first-class citizen in the secure data management arena. Moreover, we hope that cost-centric insights and architectural paradigms will fundamentally change the way systems and algorithms are designed.

References

[1] IBM 4764 PCI-X Cryptographic Coprocessor. Online at http://www-03.ibm.com/security/cryptocards/pcixcc/overview.shtml, 2007.

[2] IBM 4765 PCIe Cryptographic Coprocessor. Online at http://www-03.ibm.com/security/cryptocards/pciecc/overview.shtml, 2010.

[3] Rakesh Agrawal, Dmitri Asonov, Murat Kantarcioglu, and Yaping Li. Sovereign joins. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *ICDE*, page 26. IEEE Computer Society, 2006.

[4] Yao Chen and Radu Sion. On securing untrusted clouds with cryptography. In *WPES '10: Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 109–114, New York, NY, USA, 2010. ACM.

[5] CNN. Feds seek Google records in porn probe. Online at http://www.cnn.com, January 2006.

[6] CNN. YouTube ordered to reveal its viewers. Online at http://www.cnn.com, July 2008.

[7] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2010.

[8] Murat Kantarcioglu and Chris Clifton. Security issues in querying encrypted data. In Sushil Jajodia and Duminda Wijesekera, editors, *DBSec*, volume 3654 of *Lecture Notes in Computer Science*, pages 325–337. Springer, 2005.

[9] Luc Bouganim and Philippe Pucheral. Chip-secured data access: confidential data on untrusted server. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 131–141. VLDB Endowment, 2002.

[10] Einar Mykletun and Gene Tsudik. Incorporating a secure coprocessor in the database-as-a-service model. In *IWIA '05: Proceedings of the Innovative Architecture on Future Generation High-Performance Processors and Systems*, pages 38–44, Washington, DC, USA, 2005. IEEE Computer Society.

[11] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

[12] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.