Letter from the Special Issue Editor

The success of blockchains prompts the database community to revisit the trade-offs between security and performance in data management systems. In fact, database researchers in the past few years have made significant contributions to the understanding and advancement of blockchains. This issue focuses on systems that recently emerged (or was resurrected) at the intersection of blockchains and databases. We call them *transparent databases*, which provide data security through transparency. In particular, these systems enable the users to securely verify that both the data and its history have not been tampered with. They achieve transparency by maintaining data in an append-only ledger (a core data structure in blockchains), and protecting the ledger with authenticated data structures such as Merkle trees (core data structures in both blockchains and secure outsourced databases). Although the security community have been deploying similar systems specifically for public key infrastructure, for example, key transparency and certificate transparency, our community's interest in transparent databases stems from the challenges in building general-purpose, high-performance systems that solve real-world data management problems. This issue contains perspectives from expert researchers working on this topic. They share their views on the state-of-the-art, the use cases, and the future research directions.

The issue opens with a contribution from Henry F. Korth, in which he reminds us of how transparency is often at odds with privacy, and more importantly, their trade-offs are made for us by a trusted party. He explains how blockchains, which removes the trust on opaque institutions, can revolutionize most data-driven applications. He highlights two building blocks that are vital to such revolutions: Merkle trees and zero knowledge proofs. The former allows for selective disclosure of information, and the latter for proving correct execution without revealing the data. When combined, they enable not only integrity protection of data and computation, but also of the data provenance. The second paper, by Zhe Peng, Jianliang Xu, Haibo Hu, and Lei Chen, demonstrates how these techniques can be used to give data owners control over their data. The authors present a timely example of COVID-19 data sharing, in which users want fine-grained control of what data to share and with whom. For this use case, a Merkle tree is built over the user data and its root is published on a blockchain. To selectively share some functions on some data, the user constructs a Merkle proof for the data, and a zero-knowledge proof showing that the output is computed correctly on the input whose Merkle root is on the blockchain.

Blockchains are an important component of transparent databases, because at very least they can serve as a public bulletin board where commitments are stored. The next two papers describe the latest techniques for improving performance and security of blockchains. Junchao Chen, Suyash Gupta, Sajjad Rahnama, and Mohammad Sadoghi, present interesting insights on the advantages and limitations of two types of consensus protocols. On the one hand, Byzantine Fault Tolerance (BFT) protocols have high performance, but require strong identities, and they can be broken when the attacker steals more than f private keys. On the other hand, Proof of Work (PoW) protocols are harder to break, but they are unsustainable. The authors then propose a new protocol, called Proof of Collaboration, that aims to have the best of both worlds. Deepal Tennakoon and Vincent Gramoli discuss the state-of-the-art on blockchain sharding — the popular database technique in which the data is partitioned into multiple shards. Sharding helps scale blockchain throughputs by distributing the works. However, the key challenge in blockchain sharding, which does not exist in traditional database settings, is the presence of malicious attackers that influence shard assignments in order to insert themselves to target shards. If successful, the attackers can break the fault tolerance threshold of the target shard and subsequently break the security of the blockchain. The authors explain how *probabilistic sharding* relies on trusted sources of randomness to avoid such attacks. They propose another layer of defense, which is to make sharding transparent such that users can verify how the shard is formed.

While transparent databases can be built directly on existing blockchains that are mainly designed for cryptocurrency or assets management applications, the next paper by Dumitrel Loghin describes another approach based on *blockchain databases*. Such systems integrate blockchain and database features, and are classified into permissioned blockchains, hybrid systems, and ledger databases. They share a similar architecture that consists of a ledger storage for data history, a database storage for the states, and a broadcasting service for coordination. Hybrid systems adopt either an out-of-blockchain database design, in which the system starts with a blockchain and builds database features on top of it, or an out-of-database blockchain design, in which the system starts with a database and builds blockchain features to it. The author compares the performance of different systems and shows that ledger databases achieve the highest throughputs. The last paper, by Meihui Zhang, Cong Yue, Changhao Zhu, and Ziyue Zhong, provides in-depth analysis of ledger databases. The authors identify a number of design choices that impact the overall security and performance. They then propose a benchmarking framework, named LedgerBench, for comparing different systems. The framework contains workloads that stress the unique features of ledger databases such as verification and auditing. It has flexible APIs such that new workloads and systems can be easily added. This paper also presents interesting experimental results on their implementations of three commercial systems: LedgerDB from Alibaba, QLDB from Amazon, and SQL Ledger from Microsoft. One of the main findings is that updating the ledger and verification constitute the main performance bottleneck.

It is a real pleasure working with the authors for this issue. Their insights are refreshing and I believe the readers will learn much from reading their contributions.

Tien Tuan Anh Dinh
Singapore University of Technology and Design