**2. LITERATURE SURVEY**

**1. Guidelines on Security and Privacy in Public Cloud Computing**

**AUTHORS:** W. Jansen and T. Grance

Cloud computing can and does mean different things to different people. The common characteristics most interpretations share are on-demand scalability of highly available and reliable pooled computing resources, secure access to metered services from nearly anywhere, and displacement of data and services from inside to outside the organization. While aspects of these characteristics have been realized to a certain extent, cloud computing remains a work in progress. This publication provides an overview of the security and privacy challenges pertinent to public cloud computing and points out considerations organizations should take when outsourcing data, applications, and infrastructure to a public cloud environment.

**2. Depot: Cloud Storage with Minimal Trust**

**AUTHORS** P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish,

This article describes the design, implementation, and evaluation of Depot, a cloud storage system that minimizes trust assumptions. Depot tolerates buggy or malicious behavior by any number of clients or servers, yet it provides safety and liveness guarantees to correct clients. Depot provides these guarantees using a two-layer architecture. First, Depot ensures that the updates observed by correct nodes are consistently ordered under Fork-Join-Causal consistency (FJC). FJC is a slight weakening of causal consistency that can be both safe and live despite faulty nodes. Second, Depot implements protocols that use this consistent ordering of updates to provide other desirable consistency, staleness, durability, and recovery properties. Our evaluation suggests that the costs of these guarantees are modest and that Depot can tolerate faults and maintain good availability, latency, overhead, and staleness even when significant faults occur.

**3. Providing Database as a Service**

**AUTHORS** H. Hacigu¨mu¨ s¸, B. Iyer, and S. Mehrotra

We explore a novel paradigm for data management in which a third party service provider hosts "database as a service", providing its customers with seamless mechanisms to create, store, and access their databases at the host site. Such a model alleviates the need for organizations to purchase expensive hardware and software, deal with software upgrades, and hire professionals for administrative and maintenance tasks which are taken over by the service provider. We have developed and deployed a database service on the Internet, called NetDB2, which is in constant use. In a sense, a data management model supported by NetDB2 provides an effective mechanism for organizations to purchase data management as a service, thereby freeing them to concentrate on their core businesses. Among the primary challenges introduced by "database as a service" are the additional overhead of remote access to data, an infrastructure to guarantee data privacy, and user interface design for such a service. These issues are investigated. We identify data privacy as a particularly vital problem and propose alternative solutions based on data encryption. The paper is meant as a challenge for the database community to explore a rich set of research issues that arise in developing such a service.

**4. Fully Homomorphic Encryption Using Ideal Lattices**

**AUTHORS** C. Gentry

We propose a fully homomorphic encryption scheme -- i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. Our solution comes in three steps. First, we provide a general result -- that, to construct an encryption scheme that permits evaluation of arbitrary circuits, it suffices to construct an encryption scheme that can evaluate (slightly augmented versions of) its own decryption circuit; we call a scheme that can evaluate its (augmented) decryption circuit bootstrappable.

Next, we describe a public key encryption scheme using ideal lattices that is almost bootstrappable.

Lattice-based cryptosystems typically have decryption algorithms with low circuit complexity, often dominated by an inner product computation that is in NC1. Also, ideal lattices provide both additive and multiplicative homomorphisms (modulo a public-key ideal in a polynomial ring that is represented as a lattice), as needed to evaluate general circuits.

Unfortunately, our initial scheme is not quite bootstrappable -- i.e., the depth that the scheme can correctly evaluate can be logarithmic in the lattice dimension, just like the depth of the decryption circuit, but the latter is greater than the former. In the final step, we show how to modify the scheme to reduce the depth of the decryption circuit, and thereby obtain a bootstrappable encryption scheme, without reducing the depth that the scheme can evaluate. Abstractly, we accomplish this by enabling the encrypter to start the decryption process, leaving less work for the decrypter, much like the server leaves less work for the decrypter in a server-aided cryptosystem.

**5. Executing SQL over Encrypted Data in the Database-Service-Provider**

**Model**

**AUTHORS** H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra

Rapid advances in networking and Internet technologies have fueled the emergence of the "software as a service" model for enterprise computing. Successful examples of commercially viable software services include rent-a-spreadsheet, electronic mail services, general storage services, disaster protection services. "Database as a Service" model provides users power to create, store, modify, and retrieve data from anywhere in the world, as long as they have access to the Internet. It introduces several challenges, an important issue being data privacy. It is in this context that we specifically address the issue of data privacy.