**LITERATURE SURVEY**

**1) A fully homomorphic encryption scheme**

**AUTHORS:**  C. Gentry

We propose the first fully homomorphic encryption scheme, solving an old open problem. Such a scheme allows one to compute arbitrary functions over encrypted data without the decryption key—i.e., given encryptions E(m1), ..., E( mt) of m1, ..., mt, one can efficiently compute a compact ciphertext that encrypts f(m1, ..., mt) for any efficiently computable function f.

Fully homomorphic encryption has numerous applications. For example, it enables encrypted search engine queries—i.e., a search engine can give you a succinct encrypted answer to your (boolean) query without even knowing what your query was. It also enables searching on encrypted data; you can store your encrypted data on a remote server, and later have the server retrieve only files that (when decrypted) satisfy some boolean constraint, even though the server cannot decrypt the files on its own. More broadly, it improves the efficiency of secure multiparty computation.

In our solution, we begin by designing a somewhat homomorphic "boostrappable" encryption scheme that works when the function f is the scheme's own decryption function. We then show how, through recursive self-embedding, bootstrappable encryption gives fully homomorphic encryption.

**2) Hybrid encryption for cloud database security**

**AUTHORS:** A. Kaur and M. Bhardwaj

In cloud computing environment the new data management model is in use now a days that enables data integration and access on a large scale cloud computing as a service termed as Database-as-a-service (DAAS). Through which service provider offers customer management functionalities as well as the expensive hardware. Data privacy is the major security determinant in DAAS because data will be shared with a third party; an un-trusted server is dangerous and unsafe for the user. This paper shows a concern on the security element in cloud environment. It suggests a technique to enhance the security of cloud database. This technique provides the flexible multilevel and hybrid security. It uses RSA, Triple DES and Random Number generator algorithms as an encrypting tool.

**3) Light weight and secure database encryption using tsfs algorithm**

**AUTHORS:** D. Manivannan and R. Sujarani

Database security has paramount importance in industrial, civilian and government domains. Organizations are storing huge amount of data in database for data mining and other types of analysis. Some of this data is considered sensitive and has to be protected from disclosure. Challenges for security in database are increased due to the enormous popularity of e-business. In recent years, insider attacks gathered more attention than periodic outbreaks of malware. Database systems are usually deployed deep inside the company network and thus insiders has the easiest opportunity to attack and compromise them, and then steal the data. So data must be protected from inside attackers also. Many conventional database security systems are proposed for providing security for database, but still the sensitive data in database are vulnerable to attack because the data are stored in the form of plaintext only. Database encryption is the only solution for avoid the risk posed by this threat. This paper focuses on a security solution for protecting of data-at-rest, specifically protecting the sensitive data that resides in databases by using TSFS algorithm with three keys thus it provide more security for database. This algorithm improves the efficiency for executing the queries in database by encrypting only the sensitive data.

**4) Using in-memory encrypted databases on the cloud**

**AUTHORS:** F. Pagano and D. Pagano

Storing data in the cloud poses a number of privacy issues. A way to handle them is supporting data replication and distribution on the cloud via a local, centrally synchronized storage. In this paper we propose to use an in-memory RDBMS with row-level data encryption for granting and revoking access rights to distributed data. This type of solution is rarely adopted in conventional RDBMSs because it requires several complex steps. In this paper we focus on implementation and benchmarking of a test system, which shows that our simple yet effective solution overcomes most of the problems.

**5) A commutative encryption scheme based on ElGamal encryption**

**AUTHORS:** K.Huang and R. Tso

A commutative encryption is a kind of an encryption system that enables a plaintext to be encrypted more than once using different users' public keys. In this system, decryption is not required before the encryption/re-encryption processes. Moreover, the resulted ciphertext can be decrypted by the designated decrypters without considering the order of public keys used in the encryption/re-encryption processes. In other words, the order of keys used in encryption and in decryption do not affect the computational result. Commutative encryption scheme is found useful in many real life applications such as in secret sharing, database integration and etc. However, regardless of its usefulness, few paper demonstrates how to construct such kind of a commutative encryption. In this paper, we propose a new commutative encryption scheme based on the ElGamel encryption and provide the security proof in the random oracle model.