**CHAPTER-2**

**INTRODUCTION**

T HE invention of cloud computing has greatly eliminated the fussy tasks of managing data files by allowing clients to enjoy on-demand fast computation and massive storage resources at a very low price. Despite the conveniences, in the mechanism, clients lost physical control over their data files, which will lead to the concerns of privacy disclosure. Cryptographic techniques have been seen as a long-established approach to alleviate the concerns [1], [2], [3], which advocate that data files should be encrypted before outsourcing. As a sequence of encryption, many useful functions such as search over the outsourced data files cannot be efficiently completed. Moreover, efficient search process is indispensable for a modern cloud storage system. Searchable encryption is a cryptographic primitive that allows to execute search operations over encrypted data files, which was introduced by Song et al. [4], and can be realized in either symmetric key setting and public key setting. The former is known as symmetric searchable encryption [5], although it enjoys high efficiency in search process, it provides a terrible performance in data sharing for its complicated secret key distribution, since clients need to share the secret key which will be used for decryption when sharing an encrypted data file to others. The latter is known as public key searchable encryption [6], which is more flexible than symmetric searchable encryption at the aspect of data sharing. In public key searchable encryption, a client’s public key can be used by others to encrypt a data file shared to the client, and the client can use its secret key to generate search tokens for its queries, the server can use a search token to test whether an encrypted data file matches the query corresponding to the search token while learning nothing about the query.

**LITERATURE SURVEY**

### Title: [Practical techniques for searches on encrypted data](https://ieeexplore.ieee.org/abstract/document/848445/)

### Author: DX.Song

# Abstract: It is desirable to store data on data storage servers such as mail servers and file servers in encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching on encrypted data and provide.

### Title: [Public key encryption with keyword search](https://link.springer.com/chapter/10.1007/978-3-540-24676-3_30)

# Author: [Giuseppe Persiano](https://link.springer.com/chapter/10.1007/978-3-540-24676-3_30#auth-Giuseppe-Persiano)

# Abstract: We study the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice’s public key. An email gateway wants to test whether the email contains the keyword “urgent” so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word “urgent” is a keyword in the email without learning anything else about the email. We refer to this mechanism as Public Key Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others.

### Title: [Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack](https://ieeexplore.ieee.org/abstract/document/6296654/)

### Author:P Xu

# Abstract: Public-key encryption with keyword search (PEKS) is a versatile tool. It allows a third party knowing the search trapdoor of a keyword to search encrypted documents containing that keyword without decrypting the documents or knowing the keyword. However, it is shown that the keyword will be compromised by a malicious third party under a keyword guess attack (KGA) if the keyword space is in a polynomial size. We address this problem with a keyword privacy enhanced variant of PEKS referred to as public-key encryption with fuzzy.

### Title: [A new general framework for secure public key encryption with keyword search](https://link.springer.com/chapter/10.1007/978-3-319-19962-7_4)

# Author: P Chen

# Abstract: Public Key Encryption with Keyword Search (PEKS), introduced by Boneh et al. in *Eurocrypt’04*, allows users to search encrypted documents on an untrusted server without revealing any information. This notion is very useful in many applications and has attracted a lot of attention by the cryptographic research community. However, one limitation of all the existing PEKS schemes is that they cannot resist the Keyword Guessing Attack (KGA) launched by a malicious server. In this paper, we propose a new PEKS framework named Dual-Server Public Key Encryption with Keyword Search (DS-PEKS). This new framework can withstand all the attacks, including the KGA from the two untrusted servers, as long as they do not collude. We then present a generic construction of DS-PEKS using a new variant of the Smooth Projective Hash Functions (SPHFs), which is of independent interest.

### Title: [Practical techniques for searches on encrypted data](https://ieeexplore.ieee.org/abstract/document/848445/)

**Author**: DX Song

# Abstract: It is desirable to store data on data storage servers such as mail servers and file server sin encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching on encrypted data and provide