Data de-duplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure de-duplication in cloud storage. Although convergent encryption has been extensively adopted for secure de-duplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. This paper makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure de-duplication. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. To this end, we propose Dekey, a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments.

**Hybrid Systems**

The main goal of hybrid systems for feature selection is to extract the good characteristics of filters and wrappers and combine them in one single solution. Hybrid algorithms achieve this behavior usually by pre-evaluating the features with a filter in a way to reduce the search space to be considered by the subsequent wrapper. The term “hybrid” refers to the fact that two different evaluation methods are used, a filter-type of evaluation and the classifier accuracy evaluation methods

**2.3 IRRELEVANT FEATURES**

Irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, “good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.” Keeping these in mind, we develop a novel algorithm which can efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset. We achieve this through a new feature selection framework which composed of the two connected components of irrelevant feature removal and redundant feature elimination. The former obtains features relevant to the target concept by eliminating irrelevant ones, and the latter removes redundant features from relevant ones via choosing representatives from different feature clusters, and thus produces the final subset. The irrelevant feature removal is straightforward once the right relevance measure is defined or selected, while the redundant feature elimination is a bit of sophisticated. In our proposed FAST algorithm, it involves 1) the construction of the minimum spanning tree from a weighted complete graph; 2) the partitioning of the MST into a forest with each tree representing a cluster; and 3) the selection of representative features from the clusters.

**2.4 MINIMUM SPANNING TREE**

Although data deduplication brings a lot of benefits,security and privacy concerns arise as users’ sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different ciphertexts, making deduplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making deduplication feasible. It encrypts/decrypts a data copy with a *convergent key*, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key

and hence the same ciphertext. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted

by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform deduplication on the ciphertexts and the proof of ownership prevents the unauthorized user to access