Achieving Secure and Effective Search Services in Cloud Computing

Qin Liu[1], Shuyu Pei[1], Kang Xie[2], Jie Wu[3],
Tao Peng[4], and Guojun Wang[4]

[1] Hunan University, P. R. China
[2]Key Lab of Information Network Security,
Ministry of Public Security, P. R. China
[3]Temple University, Philadelphia, USA
[4]Guangzhou University, P. R. China

CONTENTS

01 Introduction

Introduction

- As an emerging trend, more and more data owner have begun to outsource their massive data sets to cloud servers.
- The cloud service provider (CSP) offers query services to data user.

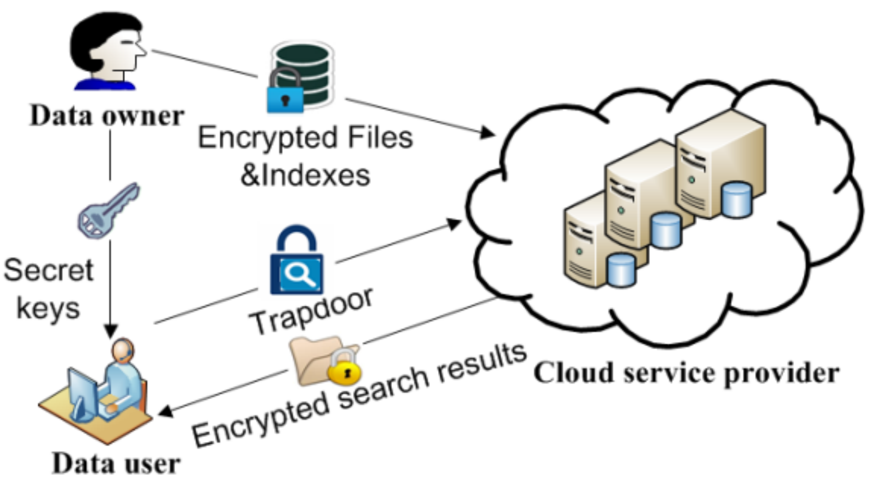


Fig. 1. System model.

Introduction

- For data security, sensitive data should be encrypted before outsourcing.
- Compared with exact search, fuzzy search allows the user to enter keywords with uncertainties or inconsistencies in their forms, and thus it can greatly improve the user experience of query services.

- Example:the user use '*' to replace several unsure letters,and issue query Q = (s *cur*ty) to retrieve appropriate files if she is unsure of the second and sixth letters of the keyword “security”.

Introduction

Research Status:

- Li's[1] scheme exploited the edit distance to quantify keyword similarity, which needs a predefined dictionary that covers possible keyword misspellings making update inefficient.

- Wang's[2] MFS scheme applied bloom filters and locality-sensitive hashing so that it has the false positive and false negative.

[1]J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. of INFOCOM, 2010.
[2]B. Wang, S. Yu, W. Lou, and Y. T. Hou,"Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in Proc. of INFOCOM, 2014.

Introduction

- In this paper, we propose a wildcard-based multikeyword fuzzy search (WMFS) scheme over the encrypted data to suport the fuzzy search.
- The main idea is to represent both the query and the index as vectors, the elements of which are set to primes or the reciprocals of primes, ensuring that all reciprocals will be eliminated only when the query matches the index.
- The level of the match can be quantified by judging whether the inner product of two encrypted vectors is an integer or not.

02 Preliminary

System Model & Adversary Model

The system is composed of the three following parts:

- Data owner
- Data user

be fully trusted

- Cloud service provider (CSP): be honest but curious.

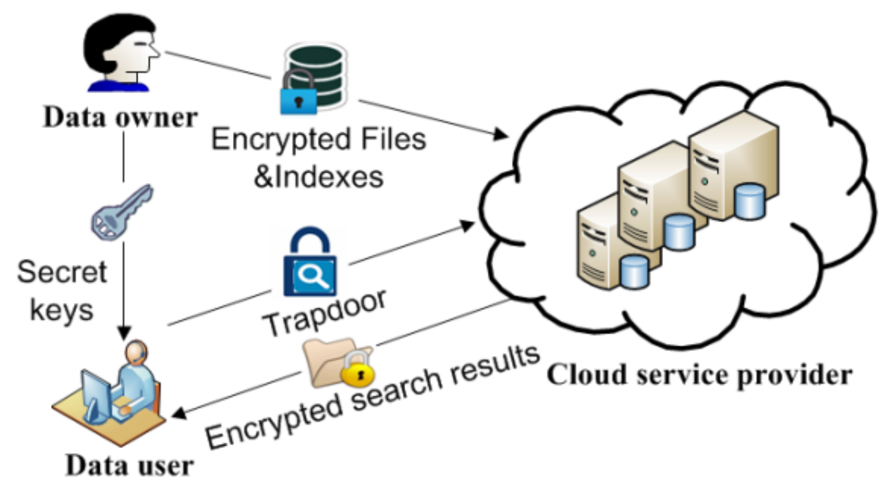


Fig. 1. System model.

KNN

KNN [3] tailored for our WMFS scheme mainly consists of the following algorithms:

- *GenKey*$(1^{\kappa}) \rightarrow sk$: generates the secret key $sk = (M_1, M_2, S)$,where M_1, M_2 are d × d invertible matrices and *S* is a bit string of d bits.

- *EncI*$(I, sk) \rightarrow I'$: It splits index vector *I* into two vectors: $\{I_a, I_b\}$, and $I' = (I_a', I_b')$ where $(I_a' = M_1^T I_a, I_b' = M_2^T I_b)$.

[3]W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in Proc. of SIGMOD, 2009.

KNN

- $EncQ(Q, sk) \rightarrow Q'$: It splits query vector Q into two vectors:(Q_a, Q_b), $Q' = (Q'_a, Q'_b) = (M_1^{-1}Q_a, M_2^{-1}Q_b)$.

- $Search(I', Q') \rightarrow v$: It calculates $v = I_a' \cdot Q'_a + I_b' \cdot Q'_b$ as the result.

$$
\begin{aligned}
& I_a' \cdot Q_a' + I_b' \cdot Q_b' \\
&= (M_1^T I_a) \cdot (M_1^{-1} Q_a) + (M_2^T I_b) \cdot (M_2^{-1} Q_b) \\
&= I_a^T Q_a + I_b^T Q_b \\
&= I^T Q
\end{aligned}
$$

03 Scheme Overview

BASIC WMFS ——Single Keyword Fuzzy Search

In basic WMFS scheme,we solve a simple problem that the user issues only single-keyword fuzzy queries to retrieve the appropriate files.

The basic WMFS scheme is constructed as follows:

- $GenKey(1^{\kappa}) \rightarrow SK$:Generate the secret keys $SK = (sk, k_f, L, P, S)$

  sk： $sk = (M_1, M_2, S)$ generated by KNN.$GenKey(1^{\kappa})$

  k_f： a κ-bit string

  L： the size of a set

  P: a set of prime numbers of L size denoted as $P=\{p_1,...,p_L\}$

  S: a set of random strings of L size denoted as $S=\{s_1,...,s_L\}$

BASIC WMFS ——Single Keyword Fuzzy Search

- BuildIndex(D, W, SK) → I :Build a searchable index I_j,a d−dimensional vector,for a keyword w_j extracted from a file D_j
  the way to caculate the vaule of $I_j[i]$ for i from 1 to d:

$$pos_l = \begin{cases} F_{k_f}(w_j(l)), & if \quad l \in [1, |w_j|] \\ F_{k_f}(\mathcal{S}[l - |w_j|]), & if \quad l \in (|w_j|, L] \end{cases} \quad (1)$$

sets $I_j[pos_l] = I_j[pos_l]/p_l$

BASIC WMFS ——Single Keyword Fuzzy Search

- EncIndex(I, SK) → I′: Encrypt the searchable index I_j into I_j' and the way to encrypt is KNN.*EncI*(*I, sk*).

BASIC WMFS ——Single Keyword Fuzzy Search

- BuildQuery(Q, SK) → Q: Build a searchable query Q,a d−dimensional vector, the way to caculate the vaule of Q[i] for i from 1 to d:

（1）if the letter is '*',the data user calculates

$pos_{l1} = F_{k_f}('a'), \ldots, pos_{l26} = F_{k_f}('z')$

and set $Q[pos_{li}] = Q[pos_{li}] \times p_l$ *for* $1 \le i \le 26$

（2）if the letter is not '*', he calculates pos_l with Eq. 1

and set $Q[pos_l] = Q[pos_l] \times p_l$

BASIC WMFS ——Single Keyword Fuzzy Search

- $\text{EncQuery}(Q, SK) \rightarrow Q'$: Generate a trapdoor Q' and the way to encrypt is KNN.$EncQ(Q, sk)$.

- $\text{Search}(I', Q') \rightarrow C_Q$: the CSP runs the KNN.$Search(I', Q')$ algorithm to calculate the inner product of I' and Q'.If the result is an integer,then the keyword corresponding file is match.

BASIC WMFS ——Single Keyword Fuzzy Search

Correctness Analysis:

Our basic WMFS scheme is considered incorrect if the following cases happen:

- Case 1. The result of $I \cdot Q$ is not an integer if query Q matches index I.
- Case 2. The result of $I \cdot Q$ is an integer if query Q mismatches index I.

Conclusion:

Case 1 and 2 are not true and our basic WMFS scheme is correct.

ADVANCED WMFS ——Multi-keyword Fuzzy Search

In the advanced WMFS scheme,it support multi- keyword fuzzy search to retrieve files of interest in one round.

The main idea is to exploit collision-free hashes to achieve constant-length vectors regardless of the number of keywords.

Compared to the basic WMFS algorithms,the advanced scheme is different from BuildIndex(D, W, SK) and EncIndex(I, SK).

ADVANCED WMFS ——Multi-keyword Fuzzy Search

- BuildIndex(D, W, SK) → I :Build a searchable index I_j,a d−dimensional vector,for keywords w_j extracted from a file D_j, and exploit collision-free hashes to caculate the vaule.
  the way to caculate the vaule of $I_j[i]$ for i from 1 to d:

$$pos_l = \begin{cases} H(j, \mathcal{I}_i[j](l)), & if \quad l \in [1, |\mathcal{I}_i[j]|] \\ H(j, \mathcal{S}[l - |\mathcal{I}_i[j]|]), & if \quad l \in (|\mathcal{I}_i[j]|, L] \end{cases} \qquad (2)$$

sets $I_j[pos_l] = I_j[pos_l] \times p_l$

ADVANCED WMFS ——Multi-keyword Fuzzy Search

- BuildQuery(Q, SK) → Q: Build a searchable query Q,a d−dimensional vector, the way to caculate the vaule of Q[i] for i from 1 to d:

（1）if the letter is '*',the data user calculates
$pos_{l1} = F_{k_f}('a'), \ldots, pos_{l26} = F_{k_f}('z')$
and set $Q[pos_l] = Q[pos_l] \times 1/p_l$ *for* 1 ≤ i ≤ 26

（2）if the letter is not '*'. he calculates pos_l with Eq. 2
and set $Q[pos_l] = Q[pos_l] \times 1/p_l$

04 Evaluation

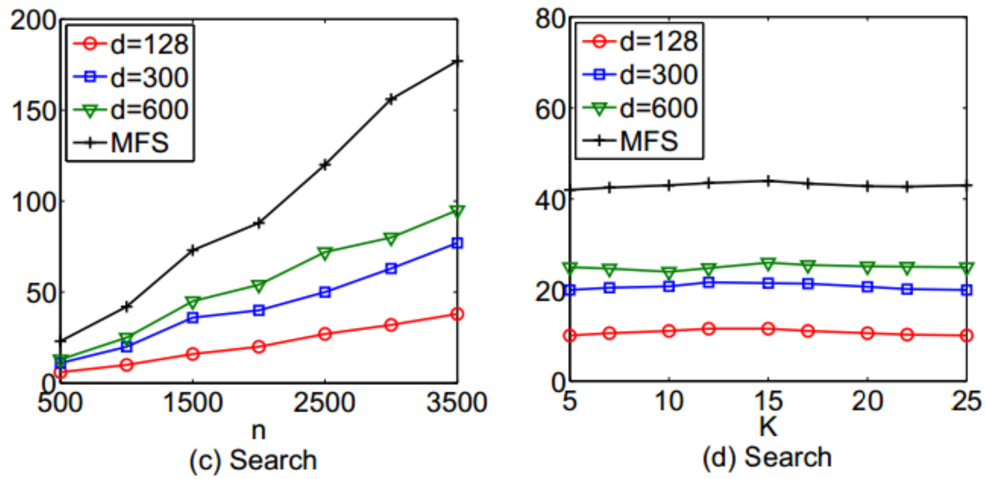
Parameter Setting

- We conduct a performance evaluation on the recent 10 years' IEEE INFOCOM publication, which includes more than 3600 files.

- The programs are implemented in Java, compiled using Eclipse 4.3.2. We apply HMAC-SHA1 as the collision-free hash function and employ the block cipher (AES) for file encryption

Computational costs

Comparison of the search time (ms) between WMFS and MFS[2].

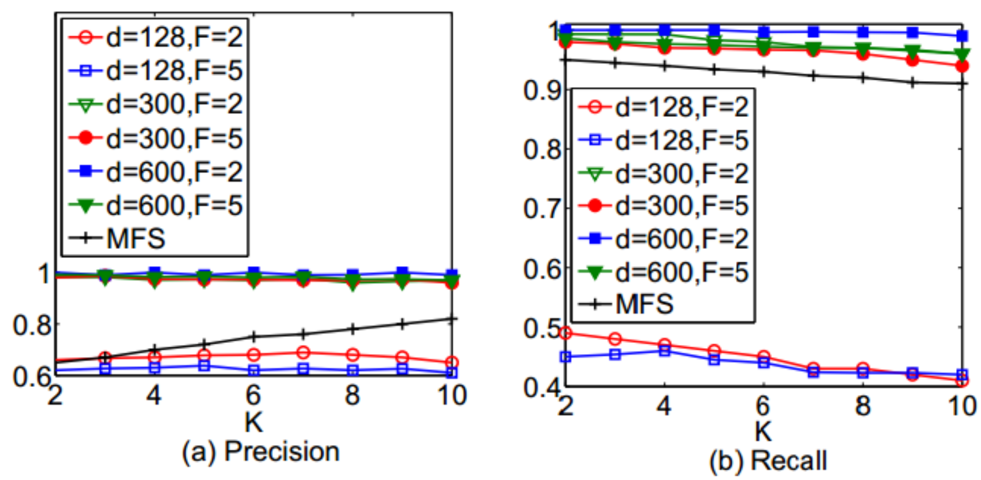


(a) The time for searching n files with fixed query keywords K = 20.
(b) The time for searching K keywords with the fixed file size n = 1000.

[2]B. Wang, S. Yu, W. Lou, and Y. T. Hou,"Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in Proc. of INFOCOM, 2014.

Precision & Recall

Comparison of accuracy between WMFS and MFS



The accuracy of our advanced WMFS scheme. The number of keywords in a query K ranges from 2 to 10.

[2]B. Wang, S. Yu, W. Lou, and Y. T. Hou,"Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in Proc. of INFOCOM, 2014.

05 Conclusion

Conclusion

- In this paper, we propose a WMFS scheme to achieve secure and effective search services in cloud computing.
- Experiment results demonstrate that our scheme is efficient and accurate.
- However, our scheme requires an order among the keywords in the multi-keyword setting.Therefore, as part of our future work, we will try to design an improved scheme supporting unordered matching

THANK YOU FOR LISTENING!